

VŠB – Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

Katedra informatiky

Mobilní aplikace pro studenty VŠB-TU Ostrava
Mobile Application for VSB-TUO Students

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Radovan Pranda**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Mobilní aplikace pro studenty VŠB-TU Ostrava**
Mobile Application for VSB-TUO Students

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce bude tvorba mobilní aplikace pro platformu Android resp. Windows Phone, využívající RESTful API, poskytnuté Centrem Informačních Technologií VŠB-TU Ostrava (CIT) k vyhledávání kontaktů v telefonním seznamu včetně informací o umístění místností a jednoduché navigace k nim včetně zobrazení polohy v mapě. Aplikace bude dále poskytovat doplňující informace pro studenty (např. rozvrh, termíny zkoušek, apod.).

1. Prostudujte existující aplikace podobného zaměření, používané v akademickém prostředí, a zjistěte současné služby poskytované CIT pro účely orientace na VŠB-TUO.
2. Zjistěte a popište současný stav API pro poskytování přístupu ke kontaktním údajům, orientaci v areálu a poskytování informací a novinek a navrhnete případná rozšíření.
3. Analyzujte a navrhnete aplikaci, zvolte vhodné implementační prostředí, popište knihovny, použité pro implementaci aplikace a jejich výběr zdůvodněte.
4. Implementujte aplikaci na platformě Android resp. Windows Phone.
5. Aplikaci otestujte alespoň na dvou různých zařízeních, a pokud to bude možné, vhodným způsobem publikujte. Výsledky testů vyhodnoťte.

Seznam doporučené odborné literatury:

- [1] Nilanchala Panigrahy: Xamarin Mobile Application Development for Android, Packt Publishing; 2. ed., 2015, 296 stran, ISBN: 978-1785280375.
- [2] Dan Hermes: Xamarin Mobile Application Development: Cross-Platform C# and Xamarin.Forms Fundamentals, 2015, 436 stran, ISBN: 978-1484202159.
- [3] Erik Hazzard: OpenLayers 2.10. Packt Publishing, 2011, 272 stran, ISBN: 978-1-84951-412-5.
- [4] Charles Petzold: Creating Mobile Apps with Xamarin.Forms, Microsoft Press, 2016, 1162 stran, [online: <https://aka.ms/xamebook>].

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

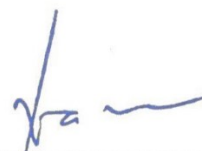
Vedoucí bakalářské práce: **Ing. Pavel Moravec, Ph.D.**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018



doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry



prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

Dňa 26.04.2018

podpis študenta

A handwritten signature in blue ink, appearing to read "Randa", written in a cursive style.

Pod'akovanie

Rád by som poďakoval Ing. Pavlovi Moravcovi, Ph.D. za odbornú pomoc a konzultáciu pri vytváraní tejto diplomovej práce a sprostredkovanie kontaktu s CIT VŠB-TU Ostrava.

Abstrakt

Táto bakalárska práca sa zaoberá vývojom univerzitnej aplikácie pre študentov VŠB-Technickej univerzity Ostrava pre operačné systémy Android a Windows Mobile. Cieľom bolo preskúmať stav existujúcich univerzitných aplikácií, vytvoriť rebríček najčastejšie sa vyskytujúcich funkcií a na základe výsledkov vytvoriť aplikáciu s využitím Rest API poskytnutých Centrom informačných technológií. Teoretická časť obsahuje štyri kapitoly zaoberajúce sa platformami Android a Windows Mobile, dostupnými nástrojmi pre medziplatformný vývoj, existujúcimi univerzitnými aplikáciami a ich funkciami, zhodnotením a popisom poskytnutých Rest API ako aj návrhom ich rozšírení. Praktická časť sa zaoberá vývojom aplikácie a jej otestovaním v praxi.

Kľúčové slová

Xamarin, Xamarin Native, medziplatformná mobilná aplikácia, Android, Windows Mobile, Rest API, študentská aplikácia

Abstract

This bachelor thesis deals with the development of the university application for students of VŠB-Technical University of Ostrava for Android and Windows Mobile operating systems. The goal was to examine the state of the existing university applications, create a list of the most frequently used features and use the results to create an application that uses the Rest API provided by the Centre for Information Technology. The theoretical part contains four chapters that deal with Android and Windows Mobile platforms, available tools for cross-platform development, existing university applications and their features, reviewing and describing the provided Rest API as well as suggestions for their extensions. The practical part deals with the development of the application and its testing in practice.

Keywords

Xamarin, cross-platform mobile application, Android, Windows Mobile, Rest API, student application

Obsah

Zoznam použitých skratiek a symbolov	9
Zoznam obrázkov.....	10
Zoznam ukážok kódu.....	11
Úvod.....	12
1 Mobilné platformy	13
1.1 Windows Phone a Windows Mobile	13
1.1.1 História.....	13
1.1.2 Verzie	13
1.2 Android.....	14
1.2.1 História.....	14
1.2.2 Verzie	14
2 Nástroje pre medziplatformný vývoj	16
2.1 Niektoré najpoužívanějšíe nástroje.....	16
2.1.1 Qt.....	16
2.1.2 Apache Cordova.....	17
2.1.3 Monocross	17
2.1.4 Sencha Touch.....	17
2.1.5 Kivy.....	17
2.2 Xamarin.....	18
2.3 Xamarin Forms a Native	18
3 Prieskum univerzitných aplikácií	19
3.1 Aplikácie pre Windows Phone (Windows Mobile).....	19
3.1.1 Niektoré dostupné univerzitné aplikácie	19
3.1.2 Výsledok prieskumu	20
3.2 Aplikácie pre Android	22
3.2.1 Niektoré dostupné univerzitné aplikácie	22
3.2.2 Aplikácie určené pre VŠB-TUO	23
3.2.3 Výsledok prieskumu	23
3.3 Porovnanie najrozšírejších funkcií v aplikáciách medzi platformami	25
4 Rest API CIT VŠB-TUO	26
4.1 Stav a popis API.....	26
4.2 Návrh rozšírení.....	26

5	Analýza a návrh aplikácie	27
5.1	Požiadavky	27
5.2	Role	27
5.3	Use-case diagram	27
5.4	Voľba vývojového nástroja	28
5.5	Návrh UI – Wireframe	29
5.5.1	Android	29
5.5.2	Windows Mobile.....	31
5.5.3	Porovnanie prostredí	31
6	Implementácia aplikácie.....	32
6.1	Návrh grafiky	32
6.2	Použité návrhové vzory	32
6.3	Použité knižnice	34
6.3.1	Zdieľané knižnice.....	35
6.3.2	Knižnice použité iba na platforme Android	36
6.4	Funkcie aplikácie.....	38
6.5	Technické požiadavky	38
7	Testovanie aplikácie.....	40
7.1	Nasadenie aplikácie a vytvorenie inšalačného balíčka.....	40
7.2	Priebeh testovania.....	40
7.2.1	Testovanie na platforme Android.....	40
7.2.2	Testovanie na platforme Windows Mobile	42
7.3	Chyby odhalené počas testovania.....	43
	Záver.....	44
	Referencie.....	45
	Obsah CD.....	46

Zoznam použitých skratiek a symbolov

RESTful	- Representational State Transfer
API	- Application Programming Interface
UWP	- Universal Windows Platform
SMS	- Short Message Service
JSON	- JavaScript Object Notation
UI	- User Interface
XML	- Extensible Markup Language
URL	- Uniform Resource Locator
URI	- Uniform Resource Identifier
CIT	- Centre for Information Technology
OS	- Operation system
PCL	- Portable class library
SAP	- Shared asset project
XAML	- Extensible Application Markup Language
VPN	- Virtual Private Network
AOSP	- Android Open Source Project
NFC	- Near field communication
Wi-Fi	- Wireless Fidelity
JIT	- Just-In-Time
JSON	- JavaScript Object Notation
QML	- Qt Modeling Language
dp, dip	- Density-independent pixel
sp, sip	- Scale-independent Pixels
apk	- Android package , resp. Android application package

Zoznam obrázkov

Obrázok 1: Príklad kódu v jazyku QML s ukážkou jeho výstupu	16
Obrázok 2: Príklad tvorby UI v jazyku Ext JS s ukážkou výsledku	17
Obrázok 3: Rozdiel medzi Xamarin Native a Xamarin Forms (spracované podľa [10]).....	18
Obrázok 4: Ukážka z aplikácie RWTHApp	19
Obrázok 5: Ukážka z aplikácie CampusApp.....	20
Obrázok 6: Graf najrozšírejších funkcií univerzitných aplikácií pre platformu Windows Phone.....	21
Obrázok 7: Ukážka z aplikácie Uplikace	23
Obrázok 8: Štatistika najrozšírejších funkcií univerzitných aplikácií pre platformu Android	24
Obrázok 9: Use-case diagram aplikácie	28
Obrázok 11: Návrh UI pre platformu Android.....	30
Obrázok 12: Návrh UI kontaktu.....	30
Obrázok 13: Návrh UI pre Windows Mobile.....	31
Obrázok 14: Ikona aplikácie Your VŠB (Tvoja VŠB).....	32
Obrázok 15: Class diagram knižnice YourVSB.....	36
Obrázok 16: Využitie dizajnových vzorov knižnice Xamarin.Android.Support	37
Obrázok 17: Využitie dizajnového vzoru CircleImageView	38
Obrázok 18: Chyby zobrazujúce sa pri probléme s pripojením k serveru	41
Obrázok 19: Zobrazenie detailu záznamu	41
Obrázok 20: Zobrazenie aplikácie na menšom displeji s väčším rozlíšením (rozloženie na šírku).....	42
Obrázok 21: Zobrazenie aplikácie na väčšom displeji s menším rozlíšením (rozloženie na šírku).....	42
Obrázok 22: Zobrazenie upozornení v prípade problémov s pripojením a neúplnosti dát.....	43
Obrázok 23: Ukážka absencie prekladov v API.....	43

Zoznam ukážok kódu

Ukážka kódu 1: Návrhový vzor Lazy Load aplikovaný na získavanie dát v sekcii Novinky	33
Ukážka kódu 2: Využitie Data bindingu v projekte pre platformu Windows Mobile	34
Ukážka kódu 3: Použitie SkiaSharp v triede ImageLoader.....	35
Ukážka kódu 4: Použitie knižnice Newtonsoft.Json na serializáciu a deserializáciu objektu.....	36

Úvod

Veľa svetových, ale aj českých vysokých škôl má vlastnú mobilnú aplikáciu uľahčujúcu život študentom a v niektorých prípadoch aj pedagógom. Preto bolo cieľom tejto bakalárskej práce vytvoriť mobilnú aplikáciu primárne určenú študentom VŠB-Technickej Univerzity Ostrava pre platformy Android a Windows Phone, využívajúcu RESTful API poskytnuté Centrom Informačných Technológií.

Výsledná aplikácia umožní užívateľovi vyhľadať kontakty v telefónnom zozname, informácie o budovách školy, vrátane zobrazenia ich polohy na mape. Okrem toho poskytuje ďalšie doplňujúce informácie pre študentov, akými sú napríklad rozvrhy, termíny skúšok a podobne.

V prvých dvoch kapitolách si v krátkosti priblížime platformy Android a Windows Mobile. Budeme sa tiež venovať nástrojom, ktoré by sme mohli využiť pri medziplatformnom vývoji aplikácie, ich zameraniu, rozdielom, funkciami a podobne.

V tretej kapitole preskúmame existujúce univerzitné aplikácie podobného zamerania a popíšeme najčastejšie nimi poskytované služby. Okrem porovnania a posúdenia, si bližšie rozoberieme pár vybraných, ktoré vynikajú oproti ostatným svojimi funkciami, dizajnom, či kompatibilitou s inými zariadeniami ako napríklad xBox, počítač alebo HoloLens.

V praktickej časti si popíšeme vývoj samotnej aplikácie a v poslednej kapitole sa budeme venovať jej testovaniu. Povieme si o chybách API, ktoré sme pri ňom odhalili, navrhujeme ich úpravy, opravy a doplnenie existujúcich API.

1 Mobilné platformy

V prvej kapitole si priblížime platformy, pre ktoré bude vyvíjaná naša aplikácia. Povieme si pár základných informácií o ich histórii a popíšeme si ich verzie, spolu s rozšíreniami, ktoré priniesli.

1.1 Windows Phone a Windows Mobile

Windows Phone je mobilný operačný systém vytvorený spoločnosťou Microsoft. Osud a ďalší vývoj tejto platformy je otázný, nakoľko bol nedávno oznámený dátum ukončenia podpory systému.

1.1.1 História

Zo zdroja [1] vieme, že bol založený na operačnom systéme Windows CE. Neskôr bolo jadro nahradené verziou Windows Phone 8 založenou na designe s názvom Metro. Veľký pokrok nastal v roku 2015 so snahou o zjednotenie s operačným systémom Windows s cieľom vytvoriť jediný plne responzívny operačný systém a bol premenovaný na Windows 10 Mobile. V roku 2017 bolo oznámené ukončenie vývoja a úplný zánik platformy a úplné ukončenie jej podpory je zatiaľ stanovený na dátum 10.12.2019¹.

1.1.2 Verzie

Windows Phone 7

Jednalo sa o prvú verziu systému, s ktorou bol predstavený nový dizajnový vzor s názvom Metro. Systém však bol výrazne oklieštený o multitasking, nakoľko neumožňoval prepínanie medzi aplikáciami.

Windows Phone 7.5

Verzia bola predstavená v roku 2011, v ktorej pribudli funkcie ako integrácia vyhľadávača Bing (webový vyhľadávač spoločnosti Microsoft) a sociálnych sietí. Jednalo sa však prevažne o zlepšenie a zvýšenie úrovne multitaskingu.

Windows Phone 8

Predstavovala veľký pokrok, nakoľko nahradila Windows CE.

Windows Phone 8.1

Vo verzii 8.1 bol systém doplnený o notifikačné centrum, možnosť nastavenia vlastného pozadia dlaždíc, podpora VPN, písanie jedným ťahom vďaka novej aplikácii, zdieľanie nastavení medzi zariadeniami, zálohovanie zariadenia a umožnila tiež presúvanie aplikácii na pamäťovú kartu. Jednou z najväčších inovácií bolo pridanie virtuálnej asistentky menom Cortana, umožňujúcej hlasové ovládanie zariadenia, vyhľadávanie na internete, ale aj spúšťať aplikácie pomocou hlasových príkazov.

Windows 10 Mobile

Bol pokusom Microsoftu o zjednotenie platforiem do jedného veľkého systému. Pokus však zlyhal, keď v roku 2017 bolo oznámené ukončenie vývoja a stanovený dátum ukončenia podpory na rok 2019.

¹ Celé vyjadrenie spoločnosti Microsoft je dostupné online na stránke:
<https://support.microsoft.com/en-us/lifecycle/search?alpha=Windows%2010%20Mobile>

Prišiel tiež s novým dizajnovým vzorom s názvom Fluent Design. Okrem nových dizajnových prvkov - napr. možnosť nastaviť priehľadnosť dlaždíc a systémového rozhrania. Boli rozšírené funkcie notifikačného centra a pridané nové nastavenia zariadenia. Časť aplikácii bola odstránená (napr. FM Rádio).

1.2 Android

Jedná sa o open source operačný systém určený pre mobilné zariadenia. Platforma je postavená na Linuxovom jadre a v súčasnej dobe sa jedná o najrýchlejšie rastúci a najrozšírenejší operačný systém v oblasti mobilných zariadení.

1.2.1 História

Systém bol vytvorený spoločnosťou Android Inc. v roku 2003 a pod názvom AOSP, tá bola v roku 2005 kúpená spoločnosťou Google Inc.

1.2.2 Verzie

Jednotlivé verzie operačného systému Android sú pomenované po rôznych typoch sladkostí. Zo zdroja [2] vieme, že nasledujúce verzie priniesli tieto významné rozšírenia:

Donut (1.6)

Jedná sa o prvú verziu vyznačujúcu sa takzvanou responzívnosťou - schopnosť systému prispôbiť sa veľkosti obrazovky.

Enclair (2.1)

Pridal možnosť pridania viacerých obrazoviek na plochu, použitie živých tapiet, ale aj funkciu Speech-to-text umožňujúcu meniť slová na text, čím nahrádza klávesnicu.

Bola tiež vytvorená navigácia Google Maps disponujúca možnosťou 3D pohľadu na mapu, zvukovej navigácie a informáciami o premávke.

Froyo (2.2)

Umožňuje na rozdiel od svojich predchodcov ovládať zariadenie pomocou hlasových príkazov, zdieľať internet pomocou Wi-Fi (tzv. Wi-Fi Hotspot). Výrazne bola zvýšená aj rýchlosť systému vďaka Dalvik JIT – softvérový komponent, ktorý analyzuje kód a aktívne ho prekladá na formu, ktorá beží rýchlejšie počas behu aplikácie.

Gingerbread (2.3)

V tejto verzii boli pridané API na tvorbu hier a podpora pre technológiu NFC. Okrem toho bol pridaný aj manažér batérie zobrazujúci štatistiky spotreby batérie aplikáciami.

Honeycomb (3.0)

S rastúcim záujmom o tablety bolo treba upraviť systém a jeho dizajn, preto boli prispôbolené veľkosti dispozície a systém sa tak stal viac tablet-friendly. Taktiež boli pridané softvérové tlačidlá, ktoré mali ukončiť éru hardvérových tlačidiel a panel rýchlych nastavení uľahčujúci prístup k základným funkciám.

Ice Cream Sandwich (4.0)

Verzia priniesla možnosť organizovania aplikácií do priečinkov, zmenu veľkostí widgetov (ovládacích prvkov prispôbených k čo najjednoduchšiemu ovládaniu alebo zobrazeniu informácií zariadenia, resp. programu), ale aj kontrolovať spotrebu mobilných dát.

Pribudla tiež služba Android Beam umožňujúca zdieľať obsah (kontakty, hudbu, obrázky, a pod.) pomocou NFC, technológie prenosu dát jednoduchým dotknutím zariadení.

Jelly Bean (4.1)

Obsahuje nového mobilného asistenta Google Now, ten poskytuje užívateľovi okrem iného informácie o počasi, novinkách, a pod. Okrem toho prichádza s interaktívnymi notifikáciami, umožňujúcimi priamo pri zobrazení notifikácie vybrať akciu, ktorá sa vykoná (napríklad v prípade zmeškaného hovoru je tu možnosť poslať SMS, zavolať späť alebo nastaviť pripomienku) a možnosťou prepínať medzi účtami rôznych užívateľov.

Kit Kat (4.4)

Vo verzii Kit Kat bol použitý nový dizajn s názvom Immersive design. Prináša tiež možnosť ovládania zariadenia hlasom – napr. posilať správy SMS, spustiť navigáciu, prehrať hudbu a pod. Evolúciou prešlo aj inteligentné vyhľadávanie, ktoré po novom začalo radieť kontakty na základe frekvencie uskutočnených volaní.

Lollipop (5.0)

Prišiel s novým dizajnovým jazykom (Material Design) kladúcim dôraz na prirodzené tieňovanie a pohyb prostredia, ale aj podporou pre hodinky, televízie (tzv. Android TV) a automobily (tzv. Android Auto).

Marshmallow (6.0)

Dáva užívateľovi možnosť pridávať a odoberať povolenia aplikáciám, čo dáva užívateľovi do rúk väčšiu moc kontrolovať aplikácie. Okrem iného výrazne predlžuje výdrž batérie a to hlavne vďaka zlepšenej optimalizácii. Bola tiež zvýšená integrácia asistenta, Google Assistant, ktorý je po novom schopný interakcie bez potreby opustiť práve otvorenú aplikáciu alebo web stránku.

Nougat (7.0)

Hlavným prínosom tejto verzie bol obrovský pokrok v multitasking, ktorý bol spôsobený zavedením multi-windows UI. Touto verziou sa tiež začína väčšia pozornosť venovať veľkosti aplikácií, spotrebe energie a RAM.

Oreo (8.0)

V tejto verzii prechádza OS Android mnohými zmenami, ktoré súvisia so zvýšením bezpečnosti a výdrže batérie. Okrem toho boli opravené mnohé chyby ako napr. chybné pridelovanie práv za behu aplikácie, či normalizácia URL. Prichádza však aj s novinkami, medzi najvýznamnejšie patrí Autofill Framework², ale aj funkcia obraz v obraze, umožňujúca prácu s viacerými oknami – napr. prácu so zariadením popri video hovore. Zmenami tiež prešli notifikácie, či cachovanie dát.

² framework poskytujúci službu automatického vyplňovania formulárov

2 Nástroje pre medziplatformný vývoj

Keďže je vývoj mobilných aplikácií časovo a finančne náročný a existujú mnohé platformy, začali vznikať nástroje umožňujúce vyvíjať aplikácie efektívnejšie a lacnejšie.

2.1 Niektoré najpoužívannejšie nástroje

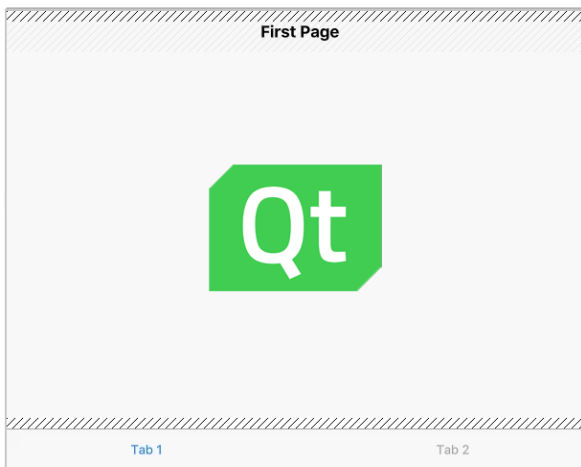
V tejto podkapitole si povieme o niektorých najpoužívannejších nástrojoch slúžiacich na medziplatformný vývoj mobilných aplikácií. Niektoré z nich však umožňujú vývoj len pre platformy Android a iOS, iné aj pre Windows Phone a Windows Mobile. Niektoré dokonca pre už mŕtve platformy ako Symbian, BlackBerry OS a ďalšie.

2.1.1 Qt³

Zo zdroja [3] vieme, že sa jedná o nástroj umožňujúci vývoj desktopových aplikácií, ale aj medziplatformných mobilných aplikácií pomocou tzv. Qt frameworku pre platformy Android, iOS, Windows Phone a Windows Mobile.

Tvorba UI prebieha pomocou Qt Creatoru jednoduchým drag&drop spôsobom alebo písaním kódu v jazyku QML – značkovací jazyk určený na tvorbu užívateľského rozhrania, sémantikou a syntaxou je podobný jazykom CSS a JSON. Metódy a back-end aplikácii sú písané v jazyku C++. Na *Obrázku 1* môžeme vidieť jednoduchý príklad použitia jazyku QML s výstupom.

Výstup



Kód

```
1 import QtQuick 2.0
2
3 App {
4     Navigation {
5         id: navigation
6         NavigationItem {
7             id: navigationItem1
8             title: "Tab1"
9             NavigationStack {
10                 id: navigationStack
11                 Page {
12                     id: page
13                     title: "First Page"
14                     AppImage {
15                         id: appImage
16                         height: 200
17                         anchors.horizontalCenter: parent.horizontalCenter
18                         anchors.verticalCenter: parent.verticalCenter
19                         source: "../assets/logo.png"
20                         fillMode: Image.PreserveAspectFit
21                     }
22                 }
23             }
24         }
25         NavigationItem {
26             id: navigationItem2
27             title: "Tab2"
28         }
29     }
30 }
```

Obrázok 1: Príklad kódu v jazyku QML s ukázkou jeho výstupu

³ Viac informácií o Qt Mobile nájdete na stránke: <https://www.qt.io/mobile-app-development/>

2.1.2 Apache Cordova

Open source nástroj umožňujúci medziplatformný vývoj aplikácií použitím jazykov HTML, CSS a JavaScript pre platformy Android, iOS, Windows Phone, BlackBerry OS, Ubuntu Touch, FirefoxOS, LGwebOS a FireOS. Sú na nej postavené mnohé frameworky⁴ ako napríklad Adobe PhoneGap, Ionic, Monaca, Onsen UI, či Cocoon.

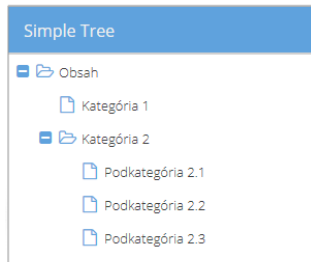
2.1.3 Monocross

Monocross je open source framework založený na frameworkoch .NET a Mono, určený na vývoj s použitím jazyka C#. Delí kód aplikácie na dve časti špecifickú pre platformu a zdieľanú časť a umožňuje tak plne medziplatformný, natívny a platformovo-špecifický vývoj dispozícií. Výsledkom vývoja sú aplikácie pre platformy: Android, iOS a Windows Phone, resp. Mobile. [4]

2.1.4 Sencha Touch

Umožňuje vývoj webových mobilných aplikácií s využitím jazyka Sencha Ext JS – JavaScript rozšírený o balíčky určenými primárne na spracovanie dát akéhokoľvek typu. Príkladom takéhoto balíčku je napríklad Ext JS Grid and Advanced Charting slúžiaci na spracovanie veľkého množstva dát do grafov. Ďalšie balíčky umožňujú vytvoriť sofistikovanú vizualizáciu dát, rýchlu analýzu robustných dát a celej aplikácii dať krásny vzhľad pomocou tém. Nástroj je teda zameraný predovšetkým na vývoj dátovo náročných webových aplikácií. [5]

Výstup



Kód

```
1 Ext.create('Ext.tree.Panel', {
2   renderTo: document.body,
3   title: 'Simple Tree',
4   width: 300,
5   height: 250,
6   root: {
7     text: 'Obsah',
8     expanded: true,
9     children: [
10      {
11        text: 'Kategória 1',
12        leaf: true
13      },
14      {
15        text: 'Kategória 2',
16        expanded: true,
17        children: [
18          { text: 'Podkategória 2.1', leaf: true },
19          { text: 'Podkategória 2.2', leaf: true },
20          { text: 'Podkategória 2.3', leaf: true }
21        ]
22      }
23    ]
24  }
25 });
```

Obrázok 2: Príklad tvorby UI v jazyku Ext JS s ukázkou výsledku

2.1.5 Kivy

Posledný z nástrojov, o ktorom si v našom prehľade povieme je Kivy.

⁴ Framework – nástroj umožňujúci vytvárať a nasadzovať aplikácie na zariadenia

Oproti ostatným nástrojom vyniká využitím jazyku Python. Vďaka spolupráci s grafickým nástrojom OpenGL ES 2 umožňuje vývoj aj graficky náročnejších aplikácií.

2.2 Xamarin

Jedná sa o nástroj umožňujúci multiplatformový vývoj mobilných aplikácií pre platformy Android, iOS, Windows Phone, Windows Mobile a vďaka kompatibilite UWP aplikácii s operačným systémom Windows aj pre počítač. Nástroj je postavený frameworku .NET a umožňuje nám tak písať kód aplikácii v jazyku C#. Umožňuje nám vyvíjať aplikácie dvomi rôznymi spôsobmi: Xamarin Forms a Xamarin Native.

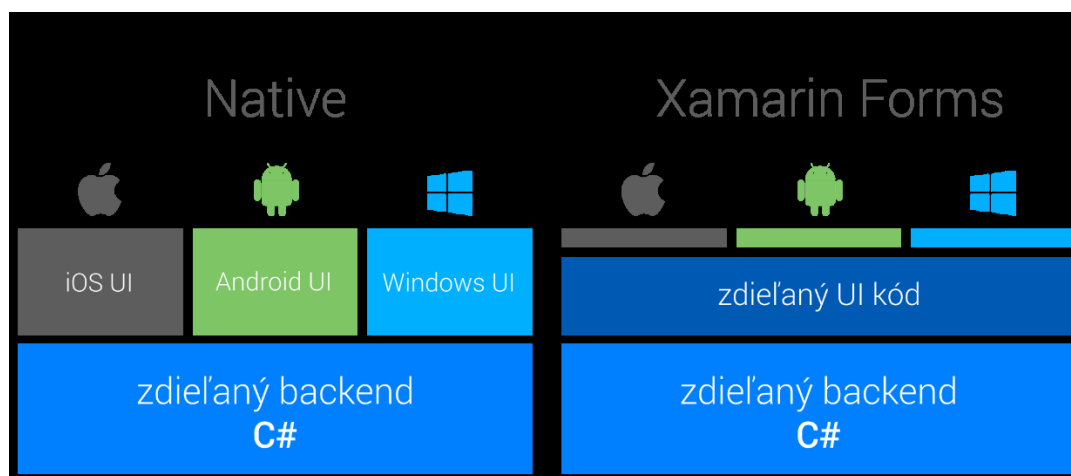
2.3 Xamarin Forms a Native

Xamarin Forms umožňuje zapísať kód zdieľaného používateľského rozhrania v jazyku C# a XAML, ktorý mapuje natívne ovládacie prvky na prvky na platformách Android, iOS a Windows Phone.

Ako môžeme vidieť na *Obrázku 3*, na rozdiel od Xamarin Forms, vývoj formou Xamarin Native vyžaduje vytvoriť pre každú platformu návrh UI samostatne. Pričom pre každú platformu musíme vytvoriť užívateľské rozhranie v jej špecifickom jazyku:

- Android – v XML
- Windows Phone a Windows Mobile – v XAML
- iOS – Xib, storyboard

Vyžaduje teda znalosť techník daných platformami. Niektoré prvky UI sú pritom dostupné len po doinštalovaní knižníc. Viac informácií sa dozvieme v podkapitole 6.3.



Obrázok 3: Rozdiel medzi Xamarin Native a Xamarin Forms (spracované podľa [10])

3 Prieskum univerzitných aplikácií

V tejto kapitole sa budeme venovať najčastejším funkciám, ktoré poskytujú univerzitné mobilné aplikácie študentom a pedagógom pre platformy Windows Phone, Windows Mobile a Android.

3.1 Aplikácie pre Windows Phone (Windows Mobile)

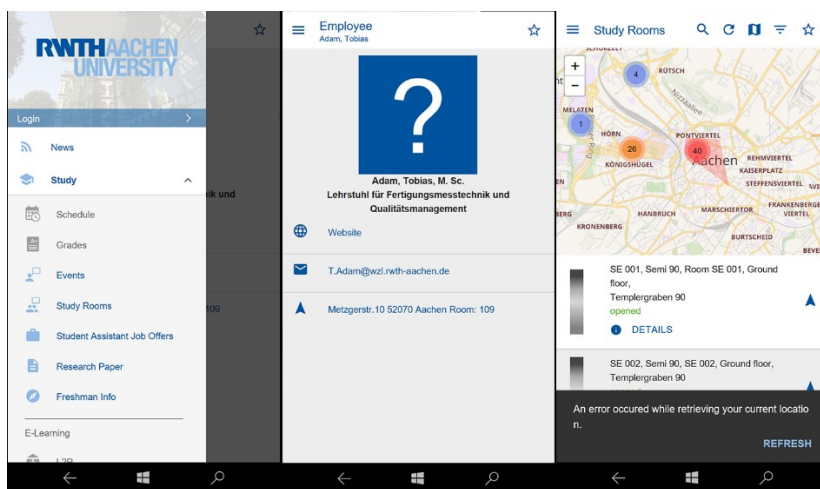
Počas vypracovávania štatistiky funkcii aplikácii sme spolu otestovali 43 aplikácii dostupných v oficiálnom obchode Microsoft Store pre platformy Windows Phone, respektíve Windows Mobile. Časť z nich vyniká svojou kompatibilitou so zariadeniami PC, HoloLens alebo Xbox. Keďže je aplikácii pre platformu Windows Phone podstatne menej ako pre platformu Android do prieskumu sme zahrnuli aj tie, ktoré patria stredným školám. Z prieskumu sme pritom vylúčili aplikácie určené iba pre počítačovú verziu systému Windows.

3.1.1 Niektoré dostupné univerzitné aplikácie

Keďže na Windows Phone ani Windows Mobile neexistuje žiadna univerzitná aplikácia určená našej univerzite, budeme svoju pozornosť venovať aplikáciám ostatných univerzít.

RWTHApp⁵

Túto aplikáciu sme si zvolili pre jej širokú škálu funkcii. Používateľom, študentom RWTH Aachen University, okrem základných informácií o dianí na univerzite poskytuje prístup do sekcie vyžadujúcich autorizáciu, akými sú napr. menza, výsledky skúšok, rozvrhy, e-learning a knižnica. Kde bolo možné si pozrieť detaily o knihách, rezervovať si ju či predĺžiť spomínanú rezerváciu a vypožičanie. Časť funkcii je dostupná verejnosti bez nutnosti autorizácie.



Obrázok 4: Ukážka z aplikácie RWTHApp

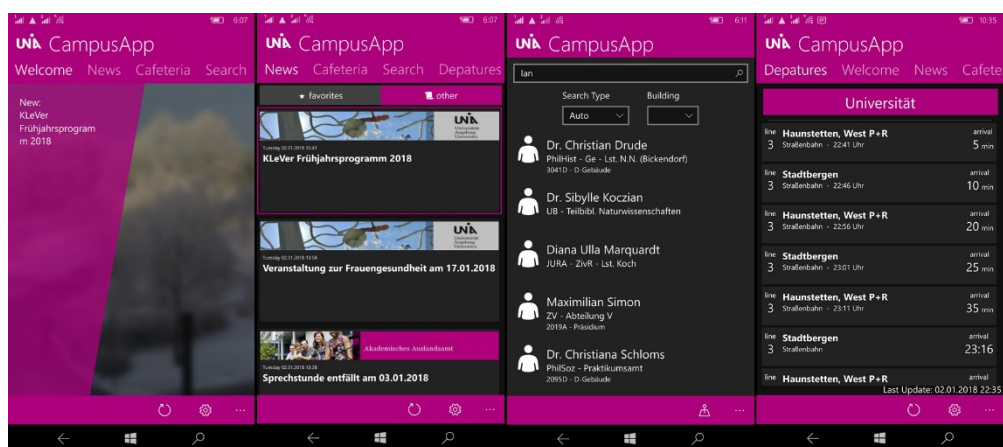
⁵ Aplikácia je dostupná na stiahnutie na:
<https://www.microsoft.com/de-de/store/p/rwthapp/9nblggh6cw3x?activetab=pivot%3aoverviewtab>

mUczelnia⁶ a Mobilna WSZiA⁷

mUczelnia a Mobilna WSZiA sú výtvorom spoločnosti PCG Academia, ktorá stojí za výtvorom aplikácií pre veľké množstvo poľských univerzít. V prípade mUczelnia sa jedná o University of Information Technology and Management in Rzeszow a Mobilna WSZiA náleží University of Management and Administration in Zamosc. Všetky ich aplikácie pritom podľa našich zistení obsahujú rovnaké funkcie. Vyznačujú sa tiež veľkou podobnosťou po dizajnovej stránke. Každá z nich obsahuje informácie o aktuálnom dianí na univerzite, výsledkoch a rozvrhu študenta. Oproti ostatným sa vyznačujú možnosťou platby, ako aj tým, že poskytujú služby učiteľom danej univerzity.

CampusApp

Je určená študentom Universität Augsburg. Disponuje základnými funkciami – informácie o aktuálnom dianí na univerzite, ponuke jedál v menze, jednoduchou navigáciou pre orientáciu v kampuse, s vyhľadávaním miestností a kontaktov. Okrem toho poskytuje informácie o spojoch smerujúcich k a od, respektíve prechádzajúcich cez areál školy. Od ostatných sa odlišuje kompatibilitou so zariadením HoloLens.



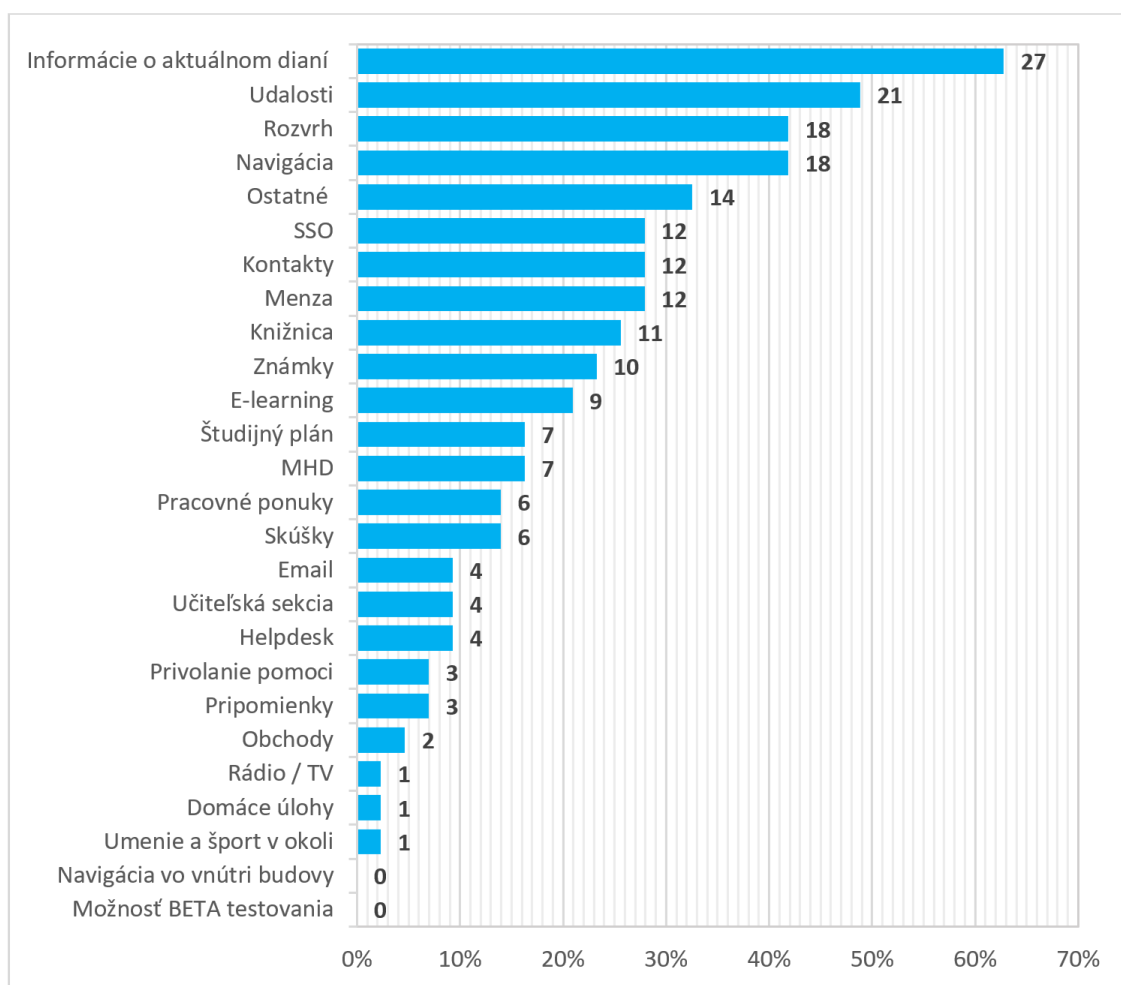
Obrázok 5: Ukážka z aplikácie CampusApp

3.1.2 Výsledok prieskumu

Pri testovaní sme zistili, že aplikácie okrem najbežnejších funkcií, akými sú napríklad informácie o aktuálnom dianí na univerzite alebo škole, plánované udalosti, navigácia, či rozvrh, obsahujú aj funkcie, ktoré sa vyskytujú v testovaných aplikáciách veľmi málo (v 1 z 43 prípadoch). Z tohto dôvodu sme ich zahrnuli do kategórie ostatné.

⁶ Aplikácia je dostupná na: <https://www.microsoft.com/de-de/store/p/muczelnia/9wzdnrcdf9nj>

⁷ Aplikácia je dostupná na: <https://www.microsoft.com/en-us/store/p/mobilna-wszia/9wzdnrcdf9nf>



Obrázok 6: Graf najrozšírenejších funkcií univerzitných aplikácií pre platformu Windows Phone

Patria k nim funkcie:

- skener QR kódov
- univerzitný chat
- počasie
- rezervácia miestností
- vybavenie preukazu ISIC
- streamovanie školskej televízie alebo rádia
- platby a stav školského konta
- živé sledovanie počtu voľných miest na parkoviskách v areáli školy
- sociálna sieť či fóra určené na komunikáciu medzi študentmi

- online rezervácia kurtov, miestností a počítačov
- online živý záznam z kamier areálu školy
- poloha bankomatov
- prístup k ubytovaciemu systému internátov

Pri porovnávaní aplikácii sme pritom zistili, že značná časť oficiálnych univerzitných aplikácii dostupných v obchode Microsoft Store má čisto informatívny charakter. Poskytujú o škole či univerzite iba základné informácie dostupné na webových stránkach školy, akými sú napríklad poloha školy na mape, informácie o založení školy, adresu a kontakt na študijné oddelenie. Väčšina dostupných oficiálnych aplikácii pritom patrí školám z Nemecka. Medzi tými, ktoré zverili vývoj externým spoločnostiam, dominovali univerzity z Poľska.

3.2 Aplikácie pre Android

Ďalšou platformou, na ktorej sme testovali aplikácie je Android. Keďže na rozdiel od platformy Windows Phone je na Android aplikácii podstatne viac, výber sme zúžili na 50 aplikácii, ktoré sú dostupné v oficiálnom obchode Play Store.

3.2.1 Niektoré dostupné univerzitné aplikácie

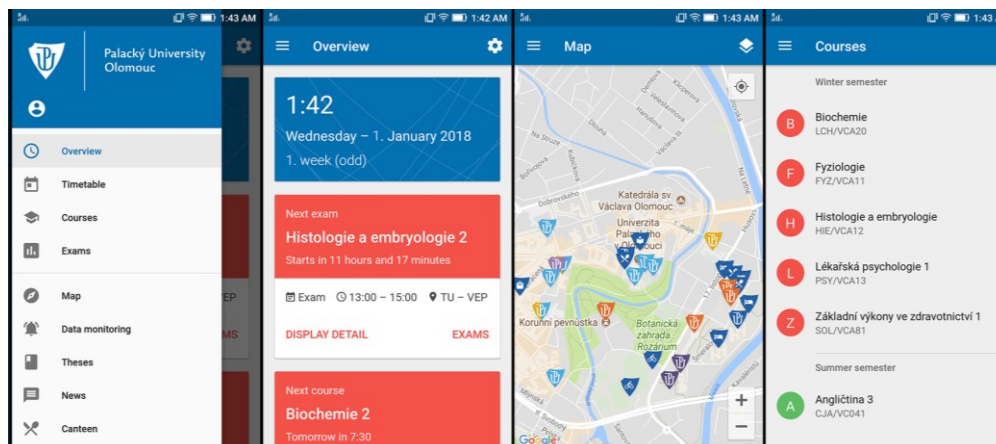
V nasledujúcej časti si rozoberieme niektoré z testovaných aplikácii, ktoré vynikali po stránke vizuálnej, kvalitatívnej alebo sa od ostatných aplikácii odlišovali niečím špeciálnym (napríklad množstvom funkcií).

AiS 2 Student

Táto slovenská aplikácia je výtvorom vývojárskeho tímu AIS2 development team - CAI PF UPJŠ, Košice, SK. Jedná sa teda o vývojársky tím patriaci univerzite. Do nášho užšieho výberu sa dostala hlavne vďaka tomu, že je aplikáciou, ktorú využíva viac ako 16 vysokých škôl z celého Slovenska. Poskytuje svojim užívateľom registráciu na skúšky, prehľad hodnotení, rozvrhov ako aj zapájať sa do diskusií v rámci školy. Všetky zo spomínaných funkcií však vyžadujú autorizáciu.

Uplikace

Uplikace je jedna z relatívne nových aplikácii, ktoré sú neustále udržiavané. Aplikácia dostáva pravidelné opravy chýb a rozšírenia. Po vizuálnej stránke vyniká čistým a prehľadným dizajnom, tzv. Material designu. Pri jej testovaní sme nenarazili na žiadne problémy. Okrem základných funkcií, ktoré sme popísali v prieskume (informácie o dani na fakulte, výsledkoch, rozvrhu a skúškach študenta, navigácia v univerzite, informácie o dani na škole) obsahovala tiež možnosť objednania jedál v menze pomocou aplikácie WebKredit, ktorej tvorcovia sú zároveň tvorcami Uplikace. Jedná sa teda o oficiálnu aplikáciu, ktorá je programovaná externou firmou pre Univerzitu Palackého v Olomouci a všetky jej funkcie vrátane informácii o dani na fakulte vyžadujú autorizáciu.



Obrázok 7: Ukážka z aplikácie Uplikace

3.2.2 Aplikácie určené pre VŠB-TUO

Rozvrh VŠB-TUO⁸

Jedná sa o študentský projekt dostupný v obchode Play Store podporujúci Android 2.3 a vyšší. Aplikácia nie je dlhšiu dobu udržiavaná a jej jedinou funkciou malo byť zobrazovanie rozvrhov.

VŠB Navi⁹

Pomáha užívateľom orientovať sa v areáli VŠB-TUO a je dostupná pre zariadenia s OS Android s verziou systému 4.0 a vyššou. Okrem navigácie pomocou mapy poskytuje tiež alternatívu v podobe navigovania kompasom, či alternatívnej reality. Ďalšou funkciou je zobrazenie noviniek z diania v škole a informácii o budovách. Aplikácia je dlhšiu dobu neudržiavaná a v niektorých bodoch nestabilná.

Mobile Guide of VŠB-TU Ostrava¹⁰

Je ďalšou aplikáciou slúžiacou užívateľom orientovať sa v areáli VŠB-TUO dostupnou pre zariadenia s OS Android s verziou systému 2.3.3 a vyššou. Okrem navigácie realizovanej pomocou Google Maps má tiež možnosť využitia alternatívnej reality. Medzi ďalšie funkcie patrí zobrazenie budov na mape a informácii o nich. Budovy sú členené do skupín na základe areálov, ktoré sú na mape farebne rozlíšené. Okrem toho obsahuje zabudovanú čítačku QR kódov, základné informácie o VŠB akými sú adresa, poloha na mape, telefónny a emailový kontakt, odkaz na oficiálnu web stránku a GPS súradnice kampusu.

3.2.3 Výsledok prieskumu

Aj keď sme testovali 50 aplikácií, počet univerzít, ktorý využíva tieto aplikácie je podstatne vyšší a to hlavne vďaka slovenskej aplikácii AiS 2 Student¹¹, ktorá slúži viac než 16-tim slovenským univerzitám.

⁸ Aplikácia je dostupná na: <https://play.google.com/store/apps/details?id=name.kocian.vsb.timetable>

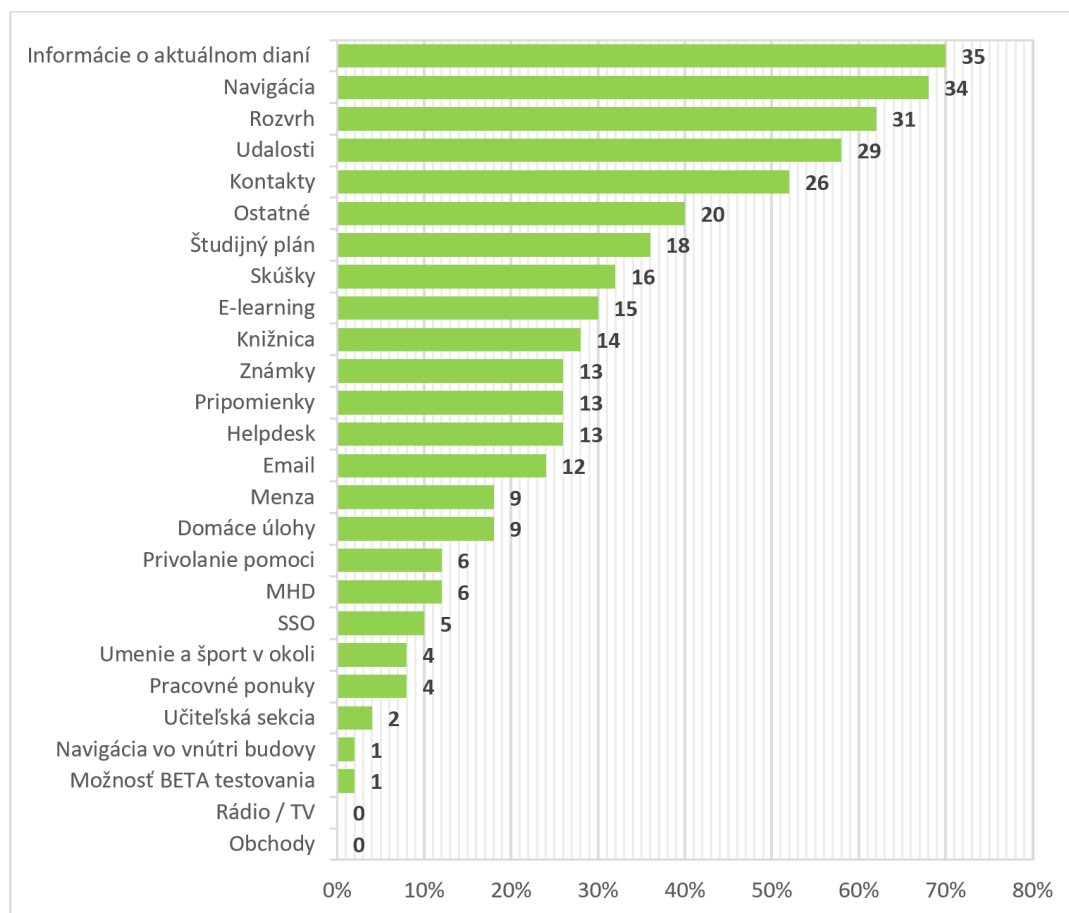
⁹ Aplikácia je dostupná na: <https://play.google.com/store/apps/details?id=cz.vsb.vsbnavi>

¹⁰ Aplikácia je dostupná na: <https://play.google.com/store/apps/details?id=com.effortix.app110>

¹¹ Aplikácia je dostupná na: <https://play.google.com/store/apps/details?id=ais.mais.android.client.termíny>

K niektorým funkciám sme nemali úplný prístup, nakoľko požadovali prihlásenie. Väčšinou sa pritom jednalo o oficiálne aplikácie.

Spomedzi aplikácií vynikala ďalej aplikácia s názvom Uplikace patriaca Univerzite Palackého v Olomouci, ktorá okrem iného ako jediná z testovaných ponúka užívateľom možnosť zapojiť sa do beta testovania.



Obrázok 8: Štatistika najrozšírenejších funkcií univerzitných aplikácií pre platformu Android

Medzi najmenej rozšírené funkcie pritom patrili obchod s univerzitnými predmetmi, výpočet priemeru a počasie. Aj tu sa však našli funkcie, ktoré sa vyskytovali u aplikácií veľmi výnimočne a boli špecifické natoľko, že sme ich pre krátkosť zahrnuli do kategórie ostatné.

Boli to:

- výpočet priemeru
- počasie
- obchody

3.3 Porovnanie najrozšírenejších funkcií v aplikáciách medzi platformami

Pri porovnaní nami vyhotovených štatistík sme prišli k záveru, že aplikácie dostupné pre platformu Android obsahujú štatisticky omnoho viac funkcií ako aplikácie dostupné na Windows a Windows Phone. Aj keď bola podpora operačného systému Windows Phone 8.1 ukončená dňa 11.07.2017[1], väčšina aplikácií je podporovaná verziou 8.1, prípadne 8.

Všeobecne štatisticky najrozšírenejšími funkciami pritom sú:

- informácie o aktuálnom dani
- navigácia v kampuse školy
- plánované udalosti
- rozvrh
- kontakty

Špecifickými funkciami u Windows aplikácii sa stali hlavne:

- streamovanie školskej televízie alebo rádia
- platby a stav školského konta

Veľkou odlišnosťou je, že majú skôr informatívny charakter a veľká časť z nich funguje ako prehliadač web stránky danej školy alebo univerzity. Takéto aplikácie sme však z nášho prieskumu vylúčili. Počet relevantných aplikácií je teda oproti Androidu razantne nižší.

Podstatným rozdielom je, že pri Androide existujú častokrát viaceré aplikácie patriace jednej univerzite a funkcie sú medzi ne rozdelené.

4 Rest API CIT VŠB-TUO

V štvrtej kapitole popíšeme stav Rest API poskytnutých oddelením CIT VŠB-TUO a navrhujeme ich rozšírenie. Popíšeme tiež chyby, s ktorými sme sa stretli počas vývoja našej mobilnej aplikácie pre študentov VŠB-TUO.

4.1 Stav a popis API

Centrum informačných technológií aktuálne poskytuje vo forme REST API služby:

- informačný servis
- kontakty
- navigácie v areáli VŠB
- prihlásenie užívateľa
- rozvrhy

Väčšina API vracia dáta vo formáte JSON. V prípade kontaktov sa tiež stretávame s formátmi XML, vCard a jCard. Niektoré časti API sú pritom neaktívne, nakoľko bolo vytvorené ich efektívnejšie riešenie. Centrum CIT pritom neustále pracuje na úprave, zmenách a skvalitňovaní služieb REST API.

U niektorých častí API sa pracuje na ich vypracovávaní a do dnešnej doby neboli sprístupnené. Jedná sa o prihlásenie užívateľa a rozvrhy.

4.2 Návrh rozšírení

V tejto podkapitole navrhujeme niektoré rozšírenia API, no keďže v súčasnej dobe nie je vo finálnej podobe, nakoľko sa stále pridávajú nové služby, je možné, že niektoré zo spomínaných funkcií budú v dohľadnej dobe pridané. Nebudeme navrhovať rozšírenia, o ktorých sme informovaní, že sa na ich tvorbe pracuje – rozvrhy a autorizácia užívateľa.

Nami navrhované rozšírenia:

- poloha tlačiarň, kioskov a miestností v budovách
- informácie o kapacite miestností s možnosťou ich rezervácie
- informácie o termínoch skúšok, priebehu semestra a študijnom pláne študenta
- základné informácie o študentovi, ktorý sa autorizuje
- prístup k dátam z knižnice (rezervácia knihy, požičané knihy, ...)

Niektoré rozšírenia nemusia byť možné z dôvodu, že jednotlivé časti bežia na rôznych systémoch, prípadne nepatria pod záštitu CIT.

5 Analýza a návrh aplikácie

V predchádzajúcich kapitolách sme vykonali prieskum dostupných univerzitných aplikácií. Ako sme zistili v súčasnej dobe existuje veľké množstvo univerzitných aplikácií. V tejto kapitole sa budeme venovať analýze nami vytváranej aplikácie.

5.1 Požiadavky

Prvým dôležitým krokom pri vývoji softvéru je ujasniť si požiadavky kladené na aplikáciu. Naším cieľom je vytvoriť aplikáciu, ktorá plne využije každú z platforiem a bude po dizajnovej stránke blízka užívateľom danej platformy. Aplikácia bude zobrazovať študentom informácie o dianí na univerzite, vyhľadávať kontakty, zobrazovať miesta patriace univerzite a po prihlásení (autorizácii) zobrazovať rozvrhy ako aj prehľad semestra.

5.2 Role

V našej aplikácii budú figurovať dva typy užívateľov, rolí.

Užívateľ

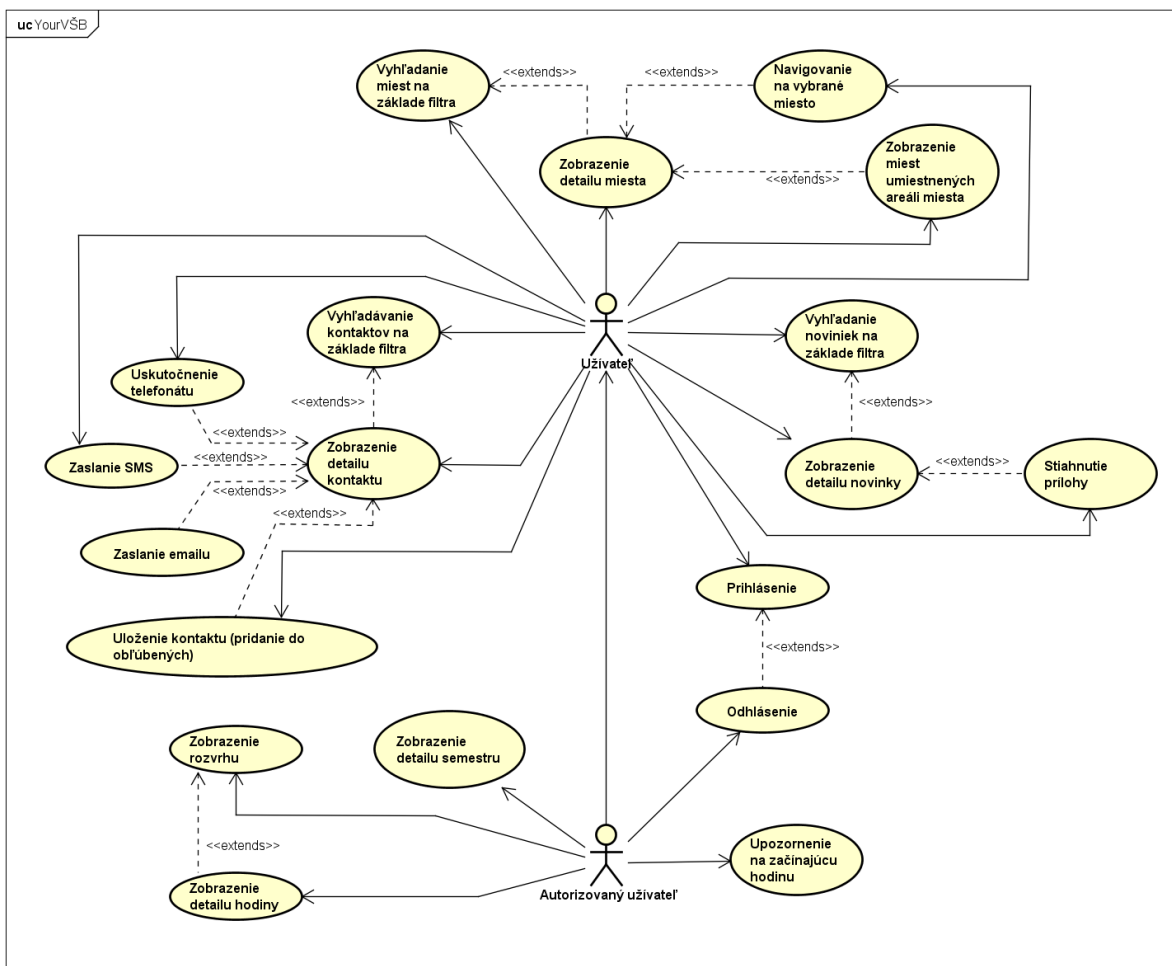
Užívateľ bude môcť prezerat' informácie o dianí na fakulte, prehliadať a vyhľadávať miesta umiestnené v areáli školy na základe filtra alebo bez jeho použitia. Vyhľadávať kontakty na základe parametrov filtra a tie si následne uložiť do obľúbených kontaktov v aplikácii.

Autorizovaný užívateľ

Autorizovaný užívateľ bude môcť navyše oproti užívateľovi prezerat' svoj rozvrh a detaily predmetov. Bude mu tiež umožnené odhlásiť sa a tým prejsť do role užívateľa.

5.3 Use-case diagram

V tejto podkapitole si detailnejšie vymedzíme role definované vyššie a funkcie obsiahnuté v aplikácii, ku ktorým budú mať prístup pomocou use-case diagramu. Niektoré funkcie z diagramu nie sú v samotnej aplikácii implementované, nakoľko boli dodané tesne pred odovzdaním tejto práce. Jedná sa o funkcie súvisiace s rozvrhmi a autorizáciou užívateľa.



Obrázok 9: Use-case diagram aplikácie

5.4 Voľba vývojového nástroja

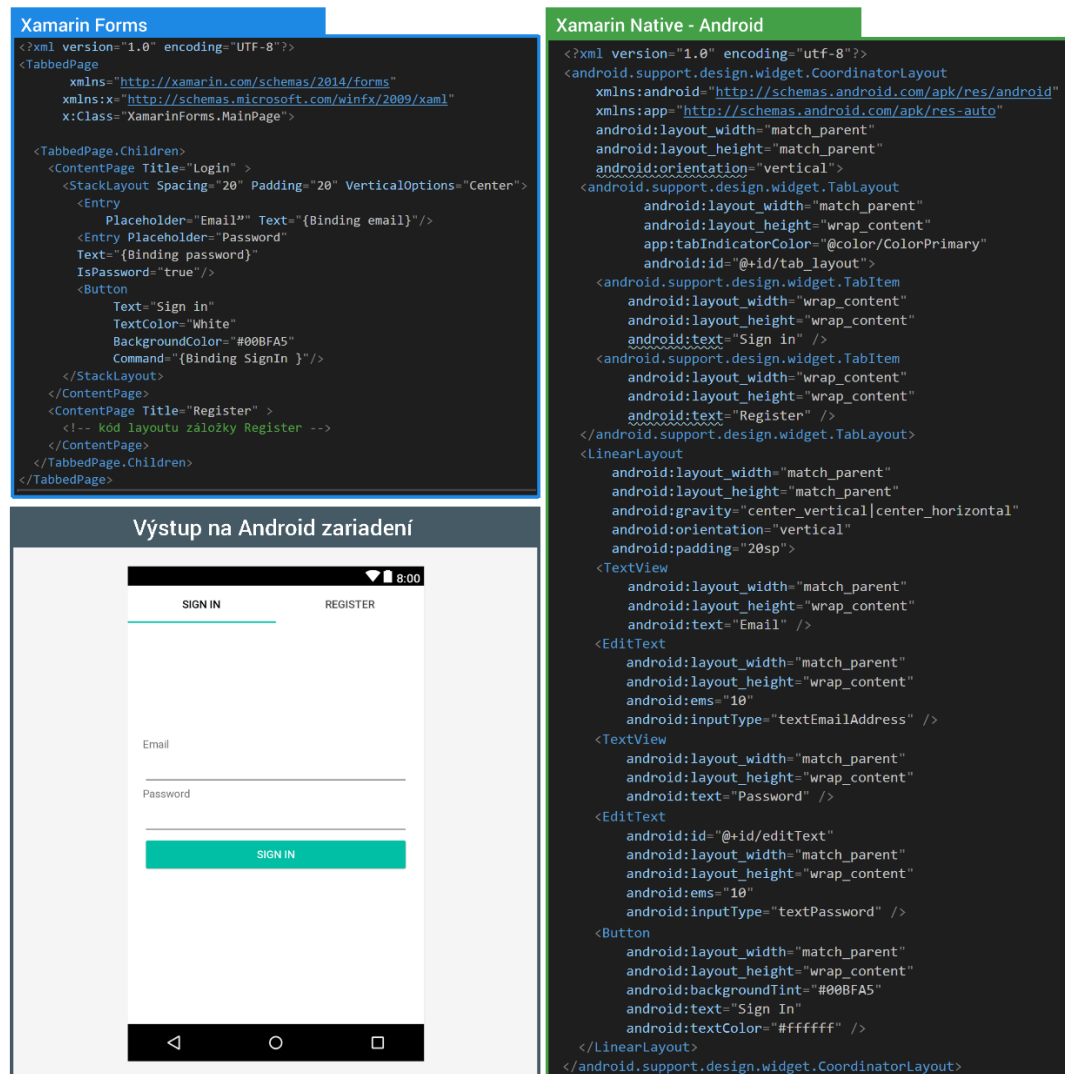
Pred začatím vývoja sme sa museli rozhodnúť, ktorý nástroj použijeme pri tvorbe našej aplikácie. Do úvahy pri tom prichádzali dve možnosti:

- vyvíjať obe aplikácie separátne (aplikácia pre platformu Android bude programovaná v jazyku Java a UWP pre Windows Phone a Windows Mobile v C# alebo C++)
- využitie nástroja pre medziplatformný vývoj

Z hľadiska praktickosti a rýchlosti vývoja sme si zvolili možnosť medziplatformného nástroja. V tomto smere sa nám ponúkalo veľké množstvo možností, akými sú napríklad Xamarin, Qt, Apache Cordova, PhoneGap, Sencha Touch, Monocross a iné. O niektorých z nich sme si povedali už v *Kapitole 2*.

Nakoniec sme sa rozhodli pre vývoj pomocou nástroja Xamarin. Aby sme platformu využili naplno a mohli vytvoriť po dizajnovej stránke aplikáciu, ktorá by bola čo najbližšia užívateľom danej platformy,

vybrali sme si formu natívneho vývoja – Xamarin Native¹². Výhodou je možnosť lepšie využiť vlastnosti každej platformy a čo najviac prispôbiť dizajn aplikácie cieľovým užívateľom.



Obrázok 10: Porovnanie tvorby UI v Xamarin Forms a Xamarin Native – Android

5.5 Návrh UI – Wireframe

V nasledujúcich podkapitolách sa budeme venovať návrhu dizajnu aplikácie.

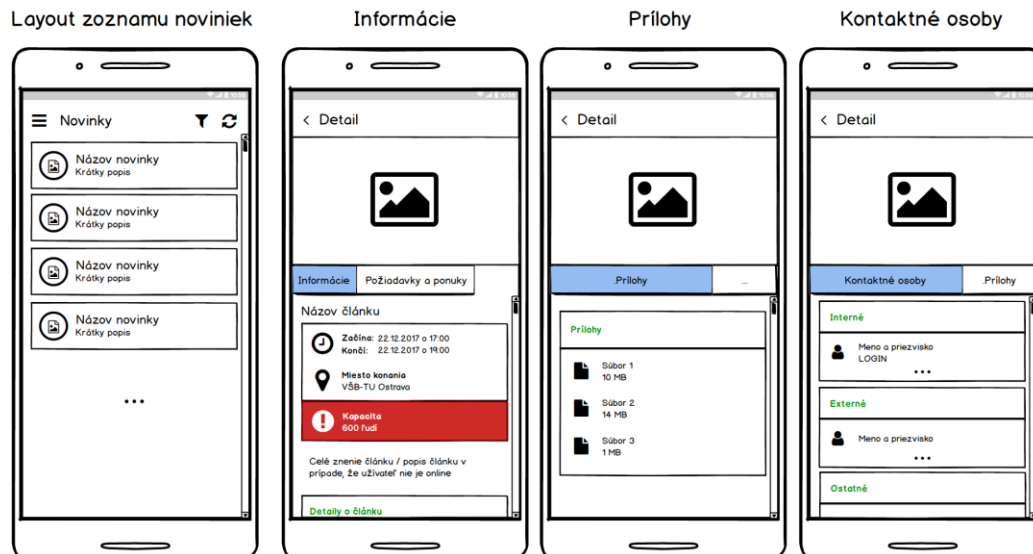
5.5.1 Android

Na Obrázku 11 môžeme vidieť niektoré návrhy vzhľadu UI pre platformu Android. Väčšina výpisov viacerých záznamov a detailov bude mať rovnaký alebo veľmi podobný vzhľad. Kladený bude dôraz na

¹² Viac informácií o Xamarin Native a rozdiely medzi Xamarin Native a Forms nájdete v Kapitole 2.32.3

využívanie dizajnových vzorov CardView a TabLayout, pričom sa budeme snažiť informácie radiť do logických kategórií, aby bolo prostredie čo najintuitívnejšie.

Layout detailu noviniek:



Obrázok 11: Návrh UI pre platformu Android

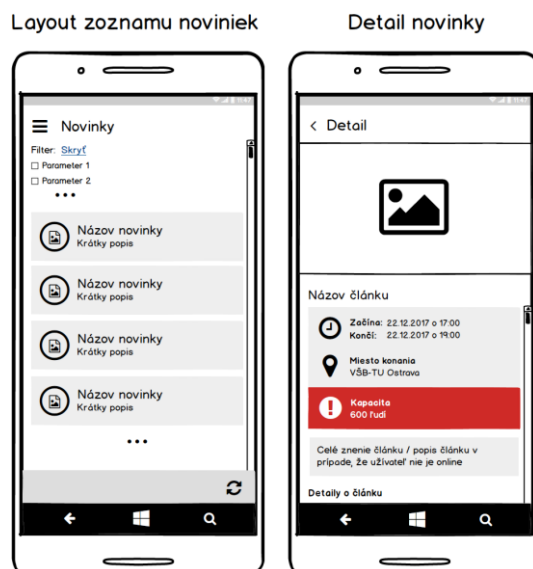
Obrázok v detaile sa bude dať skryť pri prehliadaní detailov umiestnených v spodnej časti článku. Filtrovanie v aplikácii bude realizované v prípade Androidu formou Dialogu tak, ako môžeme vidieť na Obrázku 12 vľavo.



Obrázok 12: Návrh UI kontaktu

5.5.2 Windows Mobile

Vzhľad užívateľského prostredia bude z časti podobný vzhľadu aplikácie pre Android. Pričom sa v rámci možností budeme snažiť implementovať vzhľad nového dizajnového vzoru Fluent design¹³. Aplikácia by mala byť kompatibilná s bielou aj tmavou témou Windows Mobile a prispôbiť sa tak užívateľovmu vkusu. Mieru odlišnosti od verzie aplikácie pre platformu Android môžeme vidieť na *Obrázku 13*.



Obrázok 13: Návrh UI pre Windows Mobile

5.5.3 Porovnanie prostredí

Ako môžeme vidieť, rozdiely medzi navrhovanými prostrediami sú v skutočnosti minimálne. Z implementačného hľadiska sa však jedná o veľké rozdiely. Z pohľadu dizajnu je vidieť rozdiely hlavne v logickom usporiadaní prostredia. Ako sme už vyššie spomenuli, aplikáciu sa snažíme čo najviac prispôbiť komunite využívajúcej platformu.

Zatiaľ, čo Android verzia využíva záložky na rozdelenie dát do logicky súvisiacich celkov, organizácia prvkov verzie pre Windows Mobile pôsobí blokovým usporiadaním typickým pre Fluent design. Pričom dáta sú organizované podľa priority od najdôležitejších zhora nadol. Medzi verziami je však z časti podobnosť, nakoľko u oboch OS môžeme sledovať spoločný znak v podobe akýchsi kariet, do ktorých sú dáta vkladané. V prípade Androidu sa pre tento prvok používa názov CardView.

Ďalším odlišujúcim faktorom je tiež spodná lišta obsahujúca základné operácie, ktorá je charakteristická pre platformu Windows Mobile a Windows Phone. Vďaka nej má užívateľ tie najdôležitejšie funkcie vždy poruke. V našom prípade na nej bude umiestnené tlačidlo pre aktualizovanie, obnovenie, informácií. Zatiaľ čo u druhej verzie budú tieto tlačidlá umiestnené vo vrchnej časti obrazovky.

¹³ Viac informácií o Fluent designe na <https://fluent.microsoft.com/>

6 Implementácia aplikácie

V tejto kapitole budeme rozoberať problematiku samotnej implementácie aplikácie. Aplikácia je implementovaná tak, ako je to popísané v *Kapitole 5*. Popíšeme tiež problémy, s ktorými sme sa stretli pri jej realizácii.

6.1 Návrh grafiky

Ako každá aplikácia, aj naša musí mať nejaký symbol ktorý ju reprezentuje. Ikona aplikácie je používaná vo formáte PNG. Zvyšné ikony sú v prípade Windows Mobile aplikácie súčasťou systémového vybavenia zariadenia, nakoľko využívame font Segoe MDL2 Assets.

U Androidu je princíp využívania ikon trochu zložitejší. Pre verzie s API 20 alebo nižším sa využívajú ikony vo formáte PNG, pre vyššie verzie používame vektorové ikony vo formáte XML.



Obrázok 14: Ikona aplikácie Your VŠB (Tvoja VŠB)

6.2 Použité návrhové vzory

V tejto podkapitole sa budeme venovať návrhovým vzorom použitým počas implementácie.

Lazy load

Tento návrhový vzor urýchľuje beh aplikácie. Jeho princíp totiž spočíva na odložení inicializácie objektu do bodu, dokiaľ to nie je nevyhnutné. Použili sme ho hlavne pri načítavaní noviniek tak, že pomocou asynchrónnej metódy sťahujeme iba časť informácií o článkoch a celý článok aj s obrázkami, ktoré k nemu patria, získavame až po otvorení detailu daného článku. Príklad použitia môžeme vidieť nižšie v *Ukážke kódu 1*.

```
private async Task InitData()
{
    Dictionary<string, string> Params = new Dictionary<string, string>();
    await Dispatcher.RunAsync(CoreDispatcherPriority.Normal, () =>
    {
        ProgressRingLoading.Visibility = Visibility.Visible;
        ProgressRingLoading.IsIndeterminate = true;

        /* . . . Nastavenie zdroja ListViewu . . . */
        Params.Add("brief", "true"); // nastavenie parametru brief
    });
}
```



```

        Params.Add("limit", limit.ToString()); // nastavenie počtu príspevkov
        Params.Add("lang", resources.GetString("__LangShortcut"));
    });

    All = new ObservableCollection<Report>();
    Collection<Report> AllNews = new Collection<Report>();
    connection_status = GenericBridge<Collection<Report>, Report>.Get(
        RestAPI.URLs.Base,
        RestAPI.URLs.Reports.All,
        Params,
        "infoservice",
        out AllNews);

    /* . . . Nastavenie zdroja ListViewu . . . */
    });
}

```

Ukážka kódu 1: Návrhový vzor Lazy Load aplikovaný na získavanie dát v sekcii Novinky

V prípade, že je parameter s názvom *brief* nastavený na hodnotu *true*, server vracia iba základ článku a ten teda nie je kompletný. V prípade, že je nastavený na hodnotu *false*, prípadne nie je nastavený, server zasiela celý článok. Ďalším príkladom použitia tohto návrhového vzoru je použitie parametru *limit*, ktorý obmedzuje počet príspevkov, ktoré sa sťahujú. V jeho fungovaní nám napomáha aj parameter *offset*, ktorý nám slúži na posun a stiahnutie nasledujúcich príspevkov.

Data binding

Je návrhový vzor slúžiaci na zjednodušenie vývoja GUI aplikácie prepájaním UI a bussiness logiky. V prípade správneho nastavenia sa potom zmeny premenných automaticky prenesú do GUI. Tento návrhový vzor využívame v hojnej miere hlavne v projekte pre platformu Windows Mobile. Príklad kódu môžeme vidieť nižšie na výpise *Ukážka kódu 2*.

```

<ListView x:Name="List_News" ItemsSource="{x:Bind All}"
    SelectedItem="{Binding SelectedUser}"
    SelectionChanged="ListView_SelectionChanged" >

    <ListView.ItemContainerStyle>
        <Style TargetType="ListViewItem">
            <Setter Property="Margin" Value="0, 5, 0, 0" />
            <Setter Property="Height" Value="62" />
            <Setter Property="Padding" Value="0,0, 20,0" />
            <Setter Property="Background" Value="{ThemeResource SystemAltMediumLowColor}"/>
        </Style>
    </ListView.ItemContainerStyle>

    <ListView.ItemTemplate>
        <DataTemplate x:DataType="data:Report" >
            <StackPanel Orientation="Horizontal" Padding="0" >
                <Rectangle Height="62" Width="62" HorizontalAlignment="Left" >
                    <Rectangle.Fill>

```

```

        <ImageBrush ImageSource="{Binding thumbnail.publicUrl}"
            Stretch="UniformToFill" />
    </Rectangle.Fill>
</Rectangle>
<StackPanel Margin="20, 8, 10, 8" >
    <TextBlock Text="{Binding title}" FontWeight="Bold" />
    <TextBlock Text="{Binding annotation}" />
</StackPanel>
</StackPanel>
</DataTemplate>
</ListView.ItemTemplate>
</ListView>

```

Ukážka kódu 2: Využitie Data bindingu v projekte pre platformu Windows Mobile

Adaptér

Princíp tohto návrhového vzoru spočíva v prispôbení sa objektom. Zatiaľ čo v prípade UWP aplikácie používame na zobrazenie dát Data binding, v prípade Androidu musíme pre každý ListView vytvoriť samostatný adaptér mapujúci dáta (objekty) na dispozície (layouts). Dôvodom použitia Adaptéru je nemožnosť použitia priameho Data bindingu na ListView, ako je tomu v prípade UWP aplikácie.

Okrem tohto využitia sa stretávame v našej aplikácii aj s použitím Adaptéru v prípade integrácie SQLite databázy. Adaptér nám v tomto prípade predstavuje generickú triedu obsahujúcu metódy na vytvorenie databázy alebo tabuľky, uloženie dát do lokálnej databázy, ich úpravu, či zmazanie z nej. V tomto prípade sa nám využitím adaptéru mnohonásobne zmenšilo množstvo kódu.

Foreign Key Mapper, Identity field, Row data gateway

Každý z týchto vzorov priamo súvisí s databázou a spôsobom ukladania dát v nej.

Foreign key mapper spočíva v mapovaní cudzieho kľúča na unikátny identifikátor záznamu z inej tabuľky. Pre použitie tohto návrhového vzoru sa veľmi často používa vzor Identity field. Ten je využívaný hlavne v relačných databázach ako unikátny identifikátor záznamu špecifický pre daný záznam.

Posledným zo spomenutých vzorov funguje na princípe - jeden záznam v tabuľke predstavuje jeden objekt.

Všetky vyššie spomínané vzory používame v našej SQLite lokálnej databáze, do ktorej si ukladáme dáta stiahnuté počas toho, ako je užívateľ online a následne používame v prípade, že je užívateľ off-line.

6.3 Použité knižnice

Väčšina nami použitých knižníc je kompatibilná s platformami Android, iOS aj Windows Phone. Nájdú sa medzi nimi však aj také, ktoré sú kompatibilné iba s niektorými z nich. Jedná sa o knižnice použité pri vývoji Android aplikácie, nakoľko pri vývoji aplikácie na platformu Windows Mobile sme nepoužili žiadnu knižnicu, ktorá by nebola kompatibilná s OS Android. V nasledujúcich podkapitolách sa nebudeme venovať knižniciam patriacim .NET frameworku.

6.3.1 Zdieľané knižnice

V tejto podkapitole si popíšeme knižnice, ktoré sú kompatibilné s viacerými platformami, respektíve so všetkými, ktoré Xamarin podporuje (Android, iOS, Windows Phone a Windows Mobile).

SkiaSharp

Táto knižnica slúži pre prácu s obrázkami. Umožňuje nám dáta typu `byte[]` transformovať na obrázky, ďalej meniť ich veľkosť – zmenšiť ich v prípade, že sú primalé. Jej využitie môžeme nájsť v zdieľanej časti kódu, v triede `ImageLoader`.

Jeden z príkladov využitia môžeme vidieť nižšie v *Ukážke kódu 3*. Metóda slúži ako prevodník poľa bajtov na objekt typu `SKBitmap`, ten je možné previesť na objekt typu `Bitmap` (u Androidu), alebo `Image` (u UWP), ktoré je možno následne zobraziť pomocou dizajnových vzorov určených na zobrazovanie obrázkov.

```
public static async Task<SKBitmap> loadAsync(byte[] bytes)
{
    try
    {
        var stream = new MemoryStream(bytes);
        var skbitmap = SKBitmap.Decode(stream);

        if (bytes.Length == 0 || skbitmap.Width == 0 || skbitmap.Height == 0)
            return null;
        return skbitmap;
    }
    catch
    {
        return null;
    }
}
```

Ukážka kódu 3: Použitie SkiaSharp v triede `ImageLoader`

Newtonsoft.Json

Keďže nám API vracia dáta väčšinou vo formáte JSON, je potrebné ich deserializovať. Knižnica Newtonsoft.Json nám umožňuje deserializovať dáta formátu JSON, ale aj XML. Výhodou je zároveň možnosť dáta spätne serializovať, čo využívame hlavne u platformy Android, kde serializujeme objekty a vo forme `String` do tzv. balíčkov (Bundle) slúžiacich na predávanie dát medzi Aktivitami a Fragmentmi. Príklad použitia knižnice na serializáciu a deserializáciu objektov môžeme vidieť v *Ukážke kódu 4*.

```
String JSON_SerializedItem = JsonConvert.SerializeObject(item);
// príklad serializácie objektu na formát JSON

T result = JsonConvert.DeserializeObject<T>(JSONresult);
// príklad deserializácie stringu JSON na objekt typu T
```

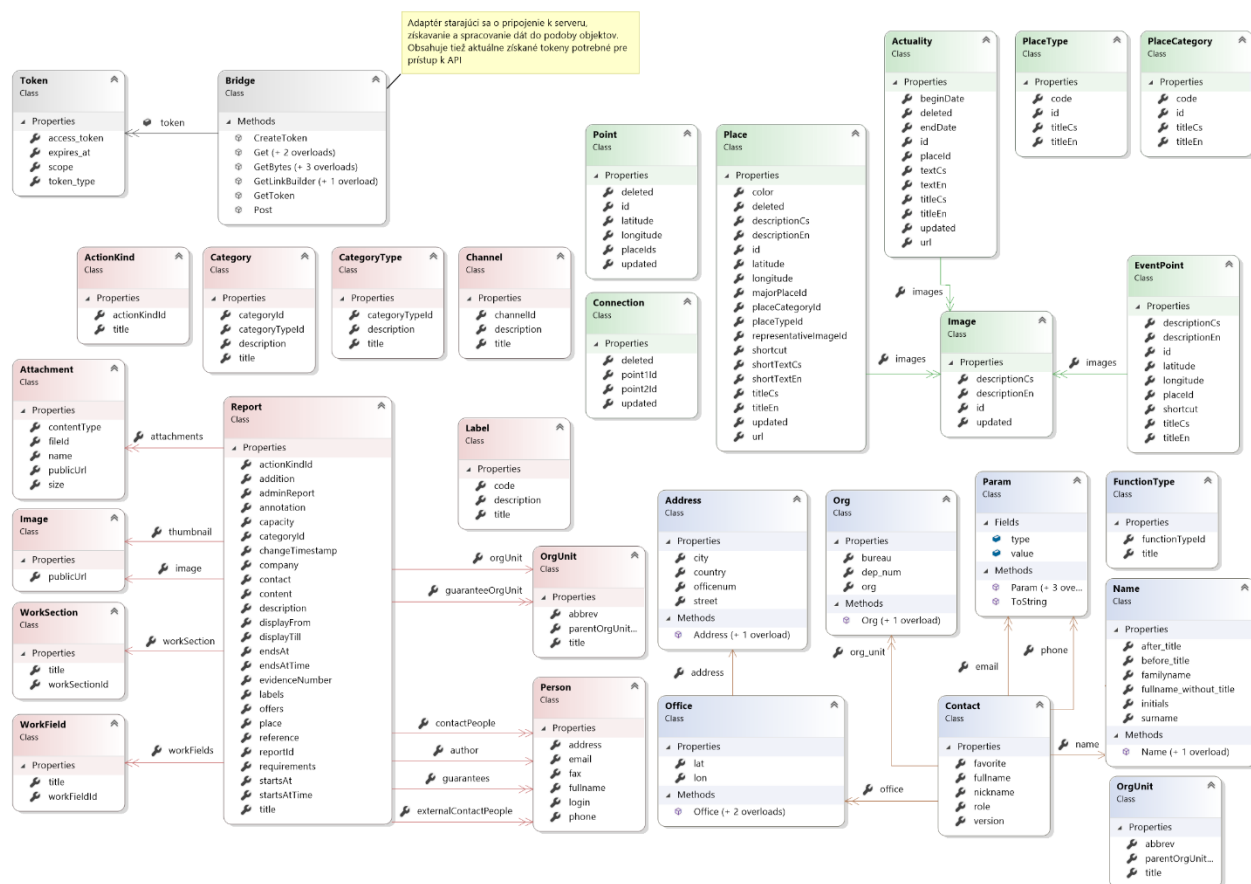
Ukážka kódu 4: Použitie knižnice Newtonsoft.Json na serializáciu a deserializáciu objektu

sqlite-pcl a sqlite-net

Knižnice nám slúžia na prístup k dátam v databáze, ich vkladanie, vyhľadávanie a získavanie z lokálnej databázy typu SQLite pomocou jednoduchých príkazov umiestnených v adaptéri *SQLiteAdapter*, ktorý je umiestnený v zdieľanej knižnici nášho projektu. Jedná sa o generickú triedu obsahujúcu metódy na vytvorenie databázy, vloženie záznamov do alebo výber z databázy.

YourVSB

Je nami vytvorená knižnica obsahujúca metódy slúžiace na získanie dát z Rest API poskytnutých CIT. Sú v nej tiež umiestnené objekty na ktoré sú tieto dáta namapované pomocou knižnice Newtonsoft.Json. Táto knižnica používa nižšie spomenuté zdieľané knižnice a predstavuje základnú funkčnú časť oboch aplikácií. Ak by sme ju chceli definovať na *Obrázku 3*, táto knižnica patrí do časti označenej ako backend aplikácie typu Xamarin.Native v jazyku C#. Štruktúru tejto knižnice, triedy, enumerátory, rozhrania a ich vzájomné vzťahy môžeme vidieť na *Obrázku 15*.



Obrázok 15: Class diagram knižnice YourVSB

6.3.2 Knižnice použité iba na platforme Android

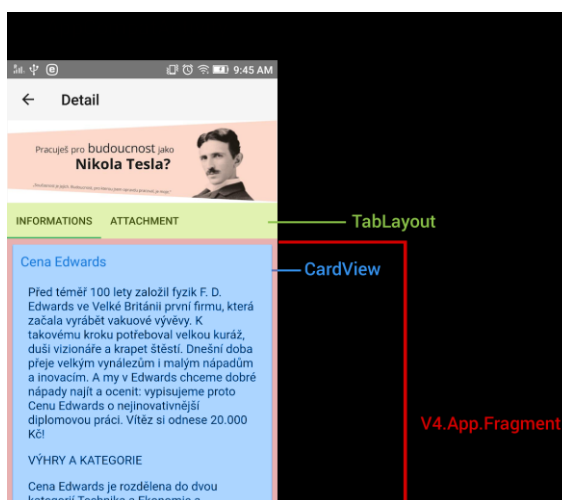
V nasledujúcej časti si rozoberieme a popíšeme knižnice použité pri vývoji aplikácie pre platformu Android. Tie však nie sú kompatibilné.

Xamarin.Android.Support

Jedná sa o knižnicu určenú pre platformu Android obsahujúcu dizajnové vzory. Využívame pritom mnohé jej časti, medzi najvýznamnejšie patria:

- Design
- Animated.Vector.Drawable
- Compat
- Constraint.Layout
- Constraint.Layout.Solver
- Fragment
- Media.Compat
- Preference
- AppCompat
- CardView
- GridLayout
- RecyclerView
- Vector.Drawable

Jednotlivé časti knižnice spolupracujú a umožňujú nám využívať prvky typické pre Material design¹⁴, akými sú napríklad CardView, TabLayout, v4.Fragment, a podobne. Niektoré z dizajnových vzorov môžeme vidieť nižšie na obrázkoch z aplikácie na *Obrázku 16* alebo nižšie v obrázkoch obrazovky priamo z aplikácie.

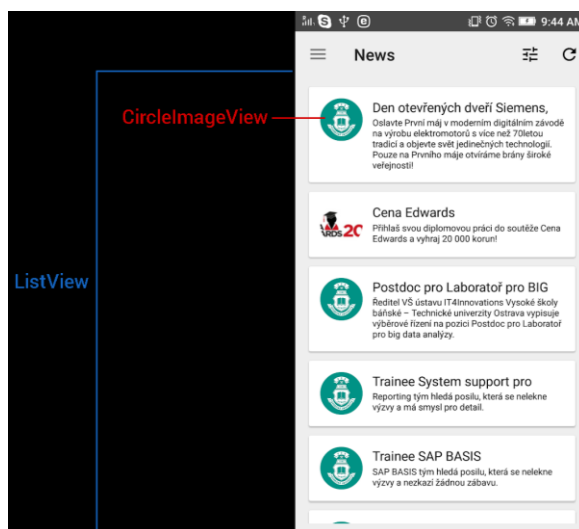


Obrázok 16: Využitie dizajnových vzorov knižnice Xamarin.Android.Support

¹⁴ Dizajnový vzor vytvorený spoločnosťou Google Inc., používaný v Android od verzie 5.0 Lollipop, založený na prirodzenom tieňovaní a pohybe prostredia. Viac informácií na: <https://material.io/>

Xamarin.Bindings.CircleImageView

Podobne ako v prípade predchádzajúcich knižníc. Aj v tomto prípade sa jedná o knižnicu implementujúcu dizajnový návrhový vzor. Ako môžeme vidieť už aj z názvu, jedná sa o CircleImageView, ktorý je používaný v ListView dispozíciách (layouts).



Obrázok 17: Využitie dizajnového vzoru CircleImageView

6.4 Funkcie aplikácie

Aplikácia poskytuje užívateľovi nasledujúce funkcie:

- prezeranie informácií o dianí na univerzite a ich zobrazenie na základe filtra
- vyhľadanie miesta v kampuse, zobrazenie informácií o ňom, jeho fotografii vrátane polohy na mape
- vyhľadávanie kontaktov na základe filtra
- zobrazenie kontaktov a ich detailov - miesta, kontaktných údajov, funkcií a podobne
- možnosť nahlásiť chybu v aplikácii
- zmeniť jazyk aplikácie

6.5 Technické požiadavky

V prípade našej aplikácie sú na zariadenia kladené následné minimálne požiadavky:

Android:

- Verzia systému: 4.4 KitKat
- Požadovaná verzia API: 20 (KitKat + Wear support)

Windows Mobile:

- Verzia systému: Windows 10 Mobile (November Update; 10.0; Build 10586)

Ako môžeme vidieť, z minimálnych požiadaviek vyplýva, že aplikácia nie je kompatibilná so zariadeniami s operačným systémom Windows Phone 8 a Windows Phone 8.1, ktorých je na trhu viac ako zariadení s Windows 10 Mobile. Je to spôsobené snahou Microsoftu o úplné ukončenie podpory pre tieto verzie platformy a nástrojom Xamarin, ktorého verzia nám znemožnila vývoj pre zariadenia s verziou systému nižšou, ako vyššie spomínaný Windows 10 Mobile (November Update; 10.0; Build 10586).

7 Testovanie aplikácie

Aplikáciu sme testovali na zariadeniach so systémom Android a Windows 10. V prípade Android zariadení sa jednalo o modely:

- Lenovo Tab 2 A7-20 s verziou Android KitKat 4.4.2, 1GB RAM, 8GB ROM
- Lenovo Vibe K5 Plus (A6020a46) s verziou Android Lollipop 5.1.1, 2GB RAM, 16GB ROM

Na testovanie UWP aplikácie sme použili zariadenia:

- Microsoft Lumia 640 XL s Windows 10 Mobile, 1GB RAM, 8GB ROM
- Microsoft Lumia 640 s Windows 10 Mobile, 1GB RAM, 8GB ROM

Pričom všetky zariadenia majú nastavený jazyk systému na Anglický (presnejšie English - US).

7.1 Nasadenie aplikácie a vytvorenie inštalačného balíčka

Nasadenie aplikácie do zariadenia na testovanie nám umožnilo priamo Visual Studio, ktoré sme pri tvorbe aplikácie využili. Pri nasadení bola najskôr preložená zdieľaná knižnica YourVSB a následne aj časť riešenia určená pre danú platformu.

Vytváranie inštalačných balíčkov prebiehalo dvoma rôznymi spôsobmi. Zatiaľ, čo vytvorenie inštalačného balíčka pre platformu Android s príponou apk prebiehalo takzvanou archiváciou a prostredie nám neponúkalo možnosť priamo prepojiť aplikáciu s predpripraveným konceptom na Play Store, v prípade aplikácie pre Windows Mobile sa nám ponúkali viaceré možnosti. Prostredie umožňuje prepojiť aplikáciu s predpripraveným konceptom aplikácie na Microsoft Store a vytvorenie balíčka určeného na pridanie aplikácie do spomínaného obchodu. Android aplikácia je dostupná na stiahnutie v Play Store na odkaze:

<https://play.google.com/apps/testing/radovanpranda.yourvsb>

Pre prístup k nej je potrebné byť členom Google+ komunity YourVŠB Alpha testers¹⁵, nakoľko je aplikácia zverejnená v režime Alpha testovania. Aplikácia pre Windows Mobile bude dostupná na stiahnutie v Microsoft Store. Kvôli dlhému procesu certifikácie zo strany Microsoftu nebolo možné pridať odkaz na ňu do tejto práce. Nájdete ho však v popise aplikácie, ktorá je uverejnená na Play Store.

7.2 Priebeh testovania

Aplikáciu sme podľa zadania otestovali na oboch platformách. Keďže sú aplikácie a chyby, s ktorými sme sa stretli totožné, v prvej časti tejto podkapitoly sa budeme venovať testovaniu na platforme Android. V druhej časti sa budeme venovať odlišnostiam Android Aplikácie oproti Windows Mobile.

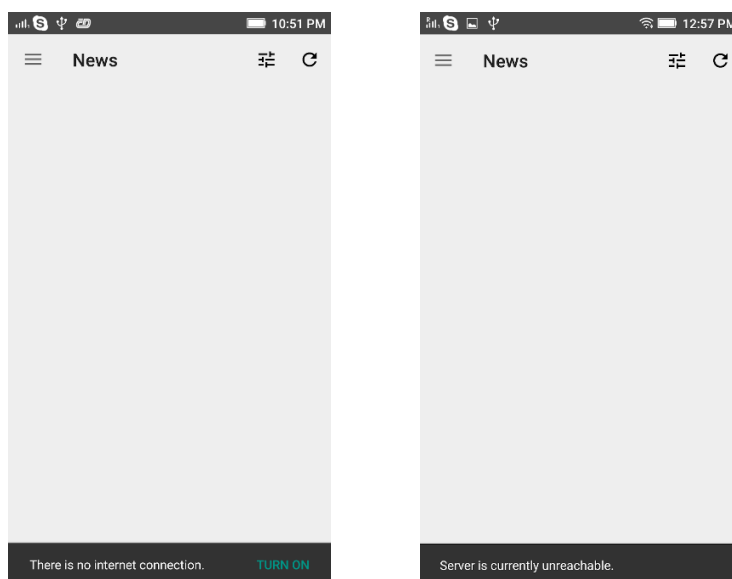
7.2.1 Testovanie na platforme Android

Po spustení aplikácie sa nám ako prvá zobrazila sekcia Novinky. Aplikácia pri prvom spustení nastavuje jazyk automaticky na základe systémového nastavenia. V našom prípade sa jazyk aplikácie nastavil na Anglický. Aplikáciu sme následne otestovali aj s nastaveným systémovým jazykom na Český. V prípade, že boli na zariadení vypnutá Wi-Fi a Mobilné dáta, ukázalo sa nám upozornenie s možnosťou zapnúť Wi-

¹⁵ O členstvo je možné požiadať na odkaze:

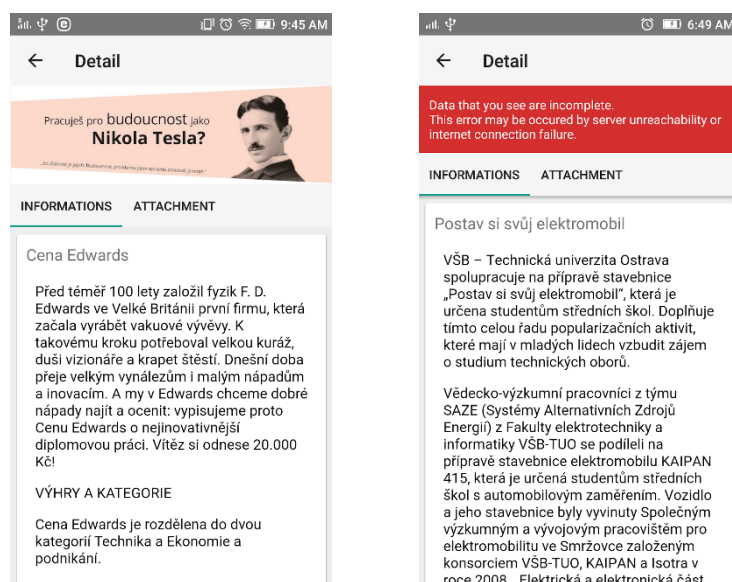
<https://plus.google.com/communities/110962681408920114324>

Fi (Obrázok 18 vľavo). V prípade, že je prijímač zapnutý, ale sťahovanie dát zo serveru CIT nebolo možné, zobrazí sa upozornenie o nedostupnosti serveru (Obrázok 18 vpravo).



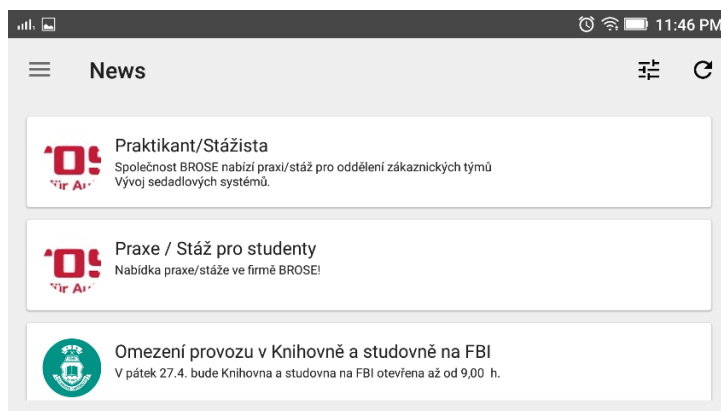
Obrázok 18: Chyby zobrazujúce sa pri probléme s pripojením k serveru

V prípade úspešného zobrazenia záznamov môžeme kliknutím na jeden z nich zobrazit' detail obsahujúci podrobnejšie informácie o ňom. Zobrazenie s podrobnosťami môžeme vidieť nižšie na Obrázku 19 vľavo. V prípade, že počas výberu položky nastane výpadok internetového pripojenia, respektíve nie je možné spojiť sa so serverom, aplikácia zobrazí upozornenie o možnej neúplnosti dát. Príklad takejto situácie možno vidieť na Obrázku 19 vpravo. Na podobnom princípe funguje aj zobrazovanie v ostatných sekciách, pričom sa líšia iba v typoch zobrazovaných dát.



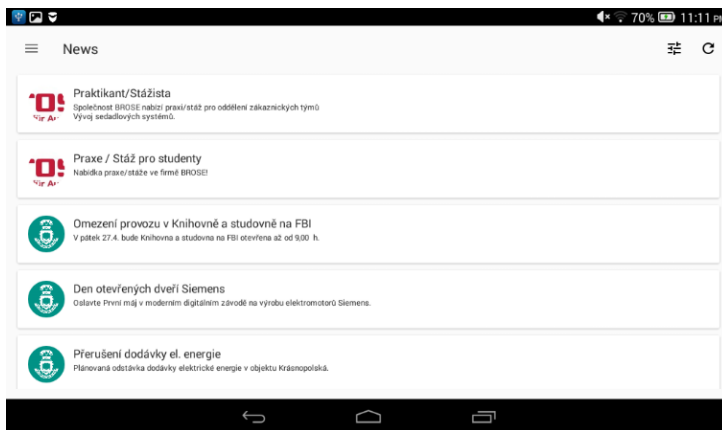
Obrázok 19: Zobrazenie detailu záznamu

V prípade, že je aplikácia spúšťaná na zariadení s väčším displejom, veľkosť písma sa prispôsobí danému zariadeniu a to vďaka použitiu jednotky sp (Scale-independent Pixels¹⁶). Rozdiel v zobrazení na displejoch s rôznymi veľkosťami a rozlíšením môžeme pozorovať na Obrázku 20 a Obrázku 21.



Obrázok 20: Zobrazenie aplikácie na menšom displeji s väčším rozlíšením (rozloženie na šírku)

Na prvom zo spomínaných obrázkov je snímok obrazovky z mobilného zariadenia s menšou fyzickou veľkosťou displeja avšak s väčším rozlíšením, zatiaľ čo na druhom môžeme vidieť snímok z tabletu, ktorý má väčšiu fyzickú veľkosť displeja avšak nižšie rozlíšenie. Ako môžeme vidieť, text je aj napriek rozdielnym fyzickým veľkostiam a výške rozlíšenia stále čitateľný a veľkosť prvkov sa prispôbila vlastnostiam zariadenia.



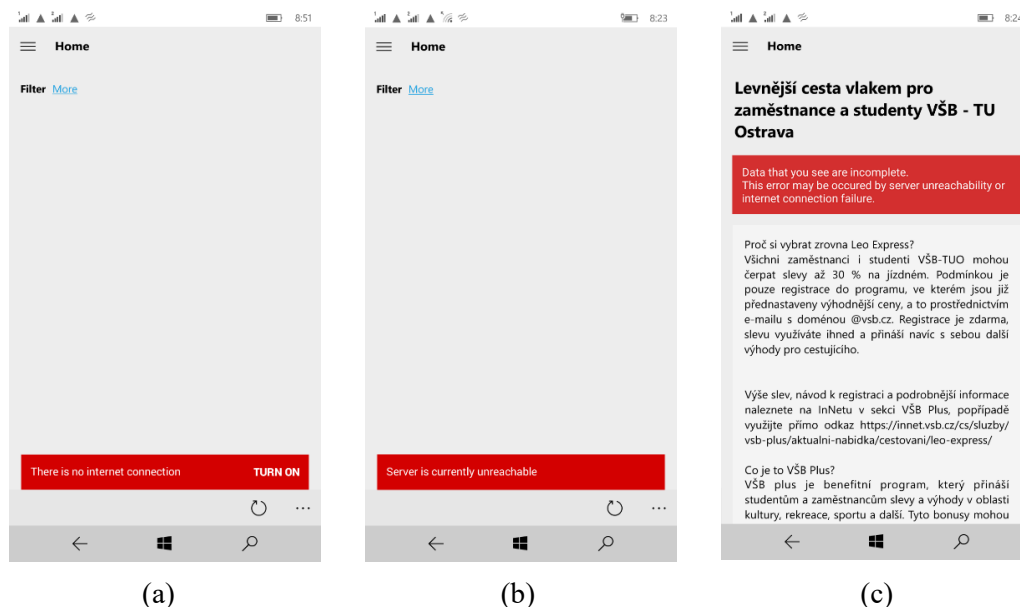
Obrázok 21: Zobrazenie aplikácie na väčšom displeji s menším rozlíšením (rozloženie na šírku)

7.2.2 Testovanie na platforme Windows Mobile

Testovanie aplikácie určenej pre Windows 10 Mobile, prebiehalo rovnako, ako môžeme vidieť v predchádzajúcej kapitole u Android aplikácie. Rozdiely sa týkali iba vizuálneho spracovania. Ako vyzerajú rozdiely spomínané vyššie môžeme vidieť na obrázkoch nižšie.

¹⁶ abstraktná jednotka, ktorá je veľmi podobná jednotke Density-independent pixel (dp, dip), od ktorého sa líši svojou závislosťou na veľkosti písma definovanej užívateľom v systémových nastaveniach

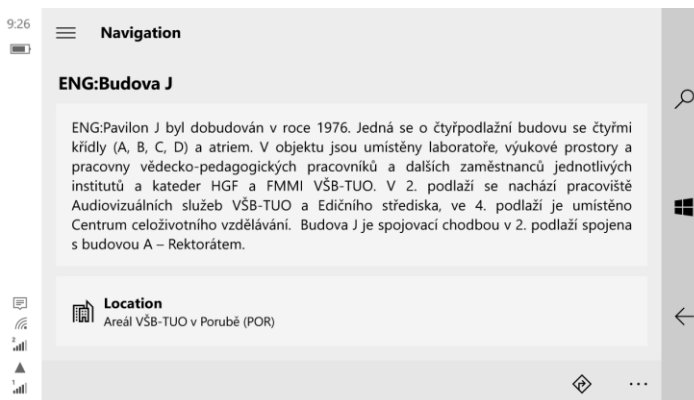
Na *Obrázku 22* môžeme vidieť tri rôzne hlásenia, ktoré sa zobrazia v prípade problémov s pripojením. Pričom časť (a) ukazuje hlásenie v prípade, že nie je užívateľ pripojený k internetu. V časti (b) tohto obrázku môžeme vidieť upozornenie na nedostupnosť serveru. Táto situácia môže nastať v prípade, že je užívateľ pripojený na internet, ale aj napriek tomu sa nebolo možné spojiť so serverom, z ktorého získava aplikácia dáta. V poslednej časti tohto obrázka (c) je zobrazenie detailu záznamu s nekompletnými dátami.



Obrázok 22: Zobrazenie upozornení v prípade problémov s pripojením a neúplnosti dát.

7.3 Chyby odhalené počas testovania

Počas testovania aplikácie sme sa na oboch platformách stretávali s drobnými chybami spôsobenými tým, že API, ktoré nám bolo poskytnuté, je momentálne vo vývoji a preto predpokladáme, že budú v dohľadnej dobe opravené. Jednou z nich je napríklad absencia prekladov (napríklad článkov v sekcii novinky, rolí v sekcii kontakty alebo miest a ich popisov v navigácii), kedy sa namiesto textov v anglickom jazyku API zobrazujú texty v češtine so značkou ENG na začiatku, niekedy bez nej. Príklad môžeme vidieť na nižšie uvedenom *Obrázku 23*.



Obrázok 23: Ukážka absencie prekladov v API

Záver

Na základe nášho prieskumu univerzitných mobilných aplikácií sme zistili, že veľké množstvo univerzít disponuje mobilnou aplikáciou určenou pre študentov, záujemcov o štúdium, prípadne aj pedagógov. Tie najčastejšie poskytujú informácie o dani na škole, chystaných udalostiach, budovách a miestnostiach v kampuse, vrátane kontaktov na pedagógov. Keďže naša škola žiadnou takouto aplikáciou nedisponuje, cieľom a motiváciou tejto bakalárskej práce bolo vytvorenie mobilnej aplikácie pre študentov VŠB-Technickej univerzity Ostrava s využitím API obdržaných od Centra informačných technológií, ktorá by im slúžila a poskytovala praktické funkcie a informácie o univerzite kedykoľvek, pohodlne a jednoducho z jedného miesta, bez nutnosti načítania stránok školy.

Na základe vykonaného prieskumu a dostupných RestAPI sme sa snažili o implementáciu tých najrozšírenejších funkcií s cieľom sprístupniť tak užívateľom, čo najväčšie množstvo informácií a funkcií. Ten nám ukázal, že medzi najrozšírenejšie funkcie univerzitných aplikácií patria informácie o aktuálnom dani, navigácia v kampuse univerzity, plánované udalosti, rozvrh a kontakty.

Z týchto funkcií nám Rest API poskytnuté CIT VŠB-TUO umožnilo implementovať a študentom tak sprístupniť informácie o dani a plánovaných udalostiach na univerzite, polohe budov v kampuse a kontakty. Okrem toho nám boli tesne pred odovzdaním tejto práce doručené aj API rozvrhov, ktoré sme však pre nedostatok času nestihli implementovať. Keďže sa jedná o prvú verziu tohto API a neustále prechádza korekciami, odstraňovaním chýb, či dopĺňovaním informácií. Počas testovania sme preto narazili na drobné chyby, ako napríklad absencia prekladov do anglického jazyka. Celkovo sú však vo veľmi dobrej kondícii a do budúcnosti by sme ocenili ich rozšírenie o ďalšie služby, ako napríklad informácie o skúškach, prihlásenom študentovi (meno, login a fakulta), či rozšírenie navigácie o polohu miestností, kioskov a tlačiarní. Ďalej by sme ocenili pridanie informácií o konzultačných hodinách do kontaktov, aby študenti vedeli, v ktorých hodinách je možné s pedagógom konzultovať diplomové práce.

Keďže v súčasnej dobe prebieha veľká rekonštrukcia webových stránok univerzity a celá réžia systému prechádza obrovskou renováciou, predpokladáme, že API bude ďalej rozširované, zdokonaľované a niektoré nami spomínané funkcie môžu byť v dohľadnej dobe pridané a sprístupnené. Naša aplikácia má teda potenciál do budúcnosti, nakoľko môže byť ďalej rozširovaná.

Pri vývoji našej aplikácie sme použili nástroj na medziplatformný vývoj Xamarin a následne ju otestovali na dvoch zariadeniach s Windows 10 Mobile a dvoch zariadeniach s operačným systémom Android verzie 5.1.1 a 4.4.2. Aplikácia bola vyvíjaná pre Windows 10 Mobile z dôvodu obmedzenia zo strany Xamarinu, ktorý už neumožňuje vývoj aplikácií pre platformy Windows Phone 8 a Windows Phone 8.1. Hlavným dôvodom je pritom snaha Microsoftu o ukončenie podpory tejto platformy.

Cieľom práce bolo vytvorenie mobilnej aplikácie pre študentov VŠB-Technickej univerzity Ostrava s využitím API obdržaných od Centra informačných technológií pre platformy Android a Windows Phone, ktorý sme vytvorením a následným publikovaním aplikácií v obchodoch Play Store a Microsoft Store naplnili.

Referencie

- [1] PC MAGAZINE ENCYCLOPEDIA. *Windows phone Definition* [online]. [cit. 2017-12-14]. Dostupné z: <https://www.pcmag.com/encyclopedia/term/61785/windows-phone>
- [2] GOOGLE LLC. *Android - History* [online]. [cit. 2017-12-16]. Dostupné z: <https://www.android.com/history>
- [3] THE QT COMPANY. *Qt for Mobile Development* [online]. [cit. 2018-01-18]. Dostupné z: <https://www.qt.io/mobile-app-development/>
- [4] MONOCROSS. *Monocross Cross-platform Mobile Development* [online]. [cit. 2018-01-19]. Dostupné z: <http://monocross.net>
- [5] SENCH. *Ext JS: JavaScript Framework for Web Apps* [online]. [cit. 2018-02-10]. Dostupné z: <https://www.sencha.com/products/extjs/#overview>
- [6] KIVY. *Kivy: Cross-platform Python Framework for NUI Development* [online]. [cit. 2018-02-10]. Dostupné z: <https://kivy.org/#home>
- [7] AIS2 DEVELOPMENT TEAM - CAI PF UPJŠ, KOŠICE, SKEDUCATION. *AiS2 student - Apps on Google Play* [online]. [cit. 2018-03-04]. Dostupné z: <https://play.google.com/store/apps/details?id=ais.mais.android.client.terminy>
- [8] MICROSOFT. *Microsoft Support* [online]. [cit. 2018-03-05]. Dostupné z: <https://support.microsoft.com/en-us/lifecycle/search?alpha=Windows%20Phone>
- [9] STOWE, Michael. *Undisturbed REST* [online]. Mulesoft, 2015, ISBN: 978-1-329-11656-6 [cit. 2018-04-17]. Dostupné z: <https://www.mulesoft.com/lp/ebook/api/restbook>
- [10] PETZOLD, Charles. *Creating Mobile Apps with Xamarin Forms* [online]. Microsoft Press, 2016 [cit. 2018-04-17]. Dostupné z: <https://aka.ms/xamebook>
- [11] MICROSOFT. *Data Binding Overview | Microsoft Docs* [online]. [cit. 2018-04-19]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/framework/wpf/data/data-binding-overview>
- [12] OODESIGN. *Adapter Pattern | Object Oriented Design* [online]. [cit. 2018-04-20]. Dostupné z: <http://www.oodeesign.com/adapter-pattern.html>
- [13] GOOGLE LLC. *Android 7.0 Behavior Changes | Android Developers* [online]. [cit. 2018-04-23]. Dostupné z: <https://developer.android.com/about/versions/nougat/android-7.0-changes.html>
- [14] GOOGLE LLC. *Android 8.0 Behavior Changes | Android Developers* [online]. [cit. 2018-04-24]. Dostupné z: <https://developer.android.com/about/versions/oreo/android-8.0-changes.html>

Obsah CD

K práci priložené CD obsahuje:

1. Text bakalárskej práce
 - PRA0100.pdf
2. Projekt so zdrojovými kódmi aplikácie
 - YourVSB
3. APK aplikácia
 - radovanpranda.yourvsb.apk
 - YourVSB.appxbundle
4. Excel dokument s tabuľkou výskumu
 - PrieskumAplikacii.xlsx